ANALYSIS OF A SCADA SYSTEM ANOMALY DETECTION MODEL BASED ON INFORMATION ENTROPY

THESIS

Jesse G. Wales, Major, USAF

AFIT-ENS-14-M-32

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-14-M-32

ANALYSIS OF A SCADA SYSTEM ANOMALY DETECTION MODEL BASED ON INFORMATION ENTROPY

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Jesse G. Wales

Major, USAF

March 2014

AFIT-ENS-14-M-32

**ANALYSIS OF A SCADA SYSTEM ANOMALY DETECTION MODEL
BASED ON INFORMATION ENTROPY**

Jesse G. Wales

Major, USAF

Approved:

//signed//   10 March 2014

_____   _____
Richard F. Deckro, DBA, (Co-Chair)   Date
Professor of Operations Research
Department of Operational Sciences

//signed//   10 March 2014

_____   _____
Jennifer L. Geffre, Maj, USAF (Co-Chair)   Date
Faculty
Department of Operational Sciences

AFIT-ENS-14-M-32

Abstract

SCADA (supervisory control and data acquisition) systems monitor and control many different types of critical infrastructure such as power, water, transportation, and pipelines. These once isolated systems are increasingly being connected to the internet to improve operations, which creates vulnerabilities to attacks. A SCADA operator receives automated alarms concerning system components operating out of normal thresholds. These alarms are susceptible to manipulation by an attacker. This research uses information theory to build an anomaly detection model that quantifies the uncertainty of the system based on alarm message frequency. Several attack scenarios are statistically analyzed for their significance including someone injecting false alarms or hiding alarms. This research evaluates the use of information theory for anomaly detection and the impact of different attack scenarios.

To My Wife and Children

Acknowledgments

I would like to express my sincere appreciation to my thesis advisors Dr. Richard Deckro and Major Jennifer Geffre. They made this thesis possible by their direction, insight, advice, and support. I am thankful for Dr. Deckro sharing his experience and knowledge of information operations throughout this effort. Major Geffre, thank you for keeping me on track and your patience. I want to thank Major Jonathan Butts and Mr. Juan Lopez for sparking my interest in SCADA system vulnerabilities. Major Brian Stone, I appreciate you sharing your design of experiments expertise.

Jesse G. Wales

Table of Contents

List of Figures

List of Tables

ANALYSIS OF SCADA SYSTEM ANOMALY DETECTION MODEL
BASED ON INFORMATION ENTROPY

I. Introduction

Background

      In the last few decades, an increasing number of supervisory control and data

acquisition (SCADA) systems have become connected to the internet, including SCADA

systems running critical infrastructure and major industries (Shaw, 2006:XVII-XVIII).

This connectivity provides easy access for operations, maintenance, and monitoring, but

creates vulnerabilities for cyber attacks. President Obama stated in Executive Order

13636, *Improving Critical Infrastructure Cybersecurity*, that the "cyber threat to critical

infrastructure continues to grow and represents one of the most serious national security

challenges we must confront" (Obama, 2013). SCADA targeted attacks, such as Stuxnet,

have shown that physical damage can occur when SCADA messages are altered and

operators are influenced.

      SCADA operators rely on alarms to direct their attention to problem areas. Tools

are used to set threshold levels and filter alarms based on operator responsibilities and the

priorities of the alarms. A balance between too many alarms and not enough alarms is

important. If the operator receives too many alarms, including false alarms (such as

when equipment is under maintenance), then they become overwhelmed or insensitive to

the alarms and miss important problems. The operator could miss significant issues with

the system if too many alarms are filtered out or the proper filters are not updated (Shaw,

2006:158-159). An operator is susceptible to an attacker changing the distribution of

alarm messages and causing the same effect. "Managing the way a SCADA system deals with alarms, and its impact on the operators, is an important consideration [and potential vulnerability]" (Shaw, 2006:156).

There are several examples of attacks on SCADA systems and accidents resulting from poor alarm management by an operator. One of the most famous attacks is Stuxnet, which targeted centrifuges in an Iranian nuclear enrichment facility (Denning, 2012:672). Stuxnet is a virus that made its way onto the Iranian SCADA system. The cyber security company Semantic published a dossier on Stuxnet where they stated the objective of Stuxnet was to "reprogram industrial control systems (ICS) by modifying code on programmable logic controllers (PLCs) to make them work in a manner the attacker intended and to hide those changes from the operator of the equipment" (Fallier, Murchu, & Chien, 2011:1). The Semantic report described several actions and attacks performed by Stuxnet that are of interest to this research: collected message traffic to observe the baseline of the system, hid and/or changed data that was sent to the operator, and changed equipment settings both overtly and covertly (Fallier *et al.*, 2011:36,47). Ultimately, Stuxnet operated undetected for over a year and may have destroyed or disrupted thousands of centrifuges and set back Iran's nuclear program by 18 months (Sanger, 2011:A1).

An example of an accident involving SCADA operators reading alarms is the Enbridge Incorporated pipeline rupture and release on July 25, 2010. According to the National Transportation Safety Board (NTSB) accident report, a ruptured pipe in Marshall, Michigan led to the release of "843,444 gallons of crude oil" that impacted wetlands, a creek, and a river in the area (National Transportation Safety Board,

2010:1,4). It took 17 hours from the time of the rupture until the control center staff was

notified of the rupture. During that time:

> Enbridge's leak detection and [SCADA] systems generated alarms consistent with
> a ruptured pipeline…the control center staff attributed the alarms to [a planned]
> shutdown and interpreted them as indications of an incompletely filled pipeline.
> (National Transportation Safety Board, 2010:xiii)

Since the operators misattributed the alarms, the operators continued to make decisions

that increased the release of oil. This accident demonstrates the SCADA operators'

reliance on alarms and the importance of responding to them correctly.

Problem Statement

According to the National Institute of Standards and Technology Guide to

Industrial Control Systems (ICS) Security, a possible SCADA vulnerability is having

"false information sent to control system operators either to disguise unauthorized

changes or to initiate inappropriate actions by system operators", (Stouffer, Falco, &

Scarfone, 2008:3.17). This research addresses two questions: 1. Can an alarm detection

model based on information theory detect message manipulation attacks? 2. What types

of attack scenarios, among those tested, significantly affect the detection model's

performance?

Research Objectives and Deliverables

There are two objectives for this research. The first objective is to build a model

for SCADA systems using information theory to detect anomalies caused by system

problems and alarm status manipulation attacks. Previous works, such as Lee and Xiang,

have shown that different information-theoretic measures can be applied in various ways

for anomaly detection (Lee & Xiang, 2001). The inputs to the model are messages from SCADA traffic and data on the alarm thresholds of the system. The model formats the data into a form that facilitates the use of information-theoretic measures. The model is used to quantify the relative information content of SCADA messages received at the human machine interface (HMI) from the remote terminal units (RTUs) using information-theoretic measures, and uses the results to determine if message frequencies are outside of normal operating conditions. The goal is to evaluate the use of this anomaly detection model on a SCADA system. This research evaluates the model on its ability to detect attacks and minimize false positive rates. The second objective of the research is the analysis of the impact of different attack scenarios on the performance of the detection model. A full factorial experiment is used to evaluate alarm manipulations including the number of alarms added, removed, and the distribution of alarm manipulations.

The deliverables include this thesis and accompanying software. The thesis provides the methodology to complete the objectives and the results from applying the model to a publicly available water treatment plant SCADA system dataset. The software deliverables include Microsoft Excel files containing the anomaly detection model and the attack scenario injection (using the data in a spreadsheet, not attacking a SCADA system).

Research Approach

The research began with a study of applications of information theory to discrete messages and anomaly detection. Information was gathered on SCADA message

protocols and possible messages that are sent to the HMI from the RTUs. Simulated data from a simple SCADA system was used to facilitate building a model to quantify the information content of SCADA messages between the RTU and HMI using message frequency and type. This model uses Excel spreadsheets and macros written by the author to perform the necessary calculations on the data, such as formatting, converting sensor data to alarm messages, logarithms, and using the expectation operator.

A selection of attack scenarios were developed to test what can or cannot be detected by the model. As discussed, the Stuxnet attack and pipeline accidents showed the devastating impact from too many false alarms and hiding real alarms. The attack scenarios varied by the number of SCADA components targeted, number of alarms added or removed, and the distribution of the manipulations. A full factorial design of experiments was used to evaluate the significance of the attack factors. This research tested which attack variables, or combinations of variables, were significant to detecting problems and the false alarm percentage.

The model was applied to an open source dataset from a water treatment facility to demonstrate its use. The procedure started with formatting the data in Excel so that the information entropy could be calculated. Then, the model was used to create a baseline for the system and set the anomaly threshold. Attack scenarios were created and applied to the dataset. They were evaluated on their significance to impact the true positive and false positive detection percentages. The attack scenarios covered a range of targets, severity, and changing rates. The statistical analysis of the data showed how information entropy might be used to detect changes in a SCADA system due to an attack using message manipulation. The results provide insight into the effectiveness of information

entropy for anomaly detection on a SCADA system and the significance of different types of attack scenarios. The research framework shown in Figure 1 is a summary of the research approach.
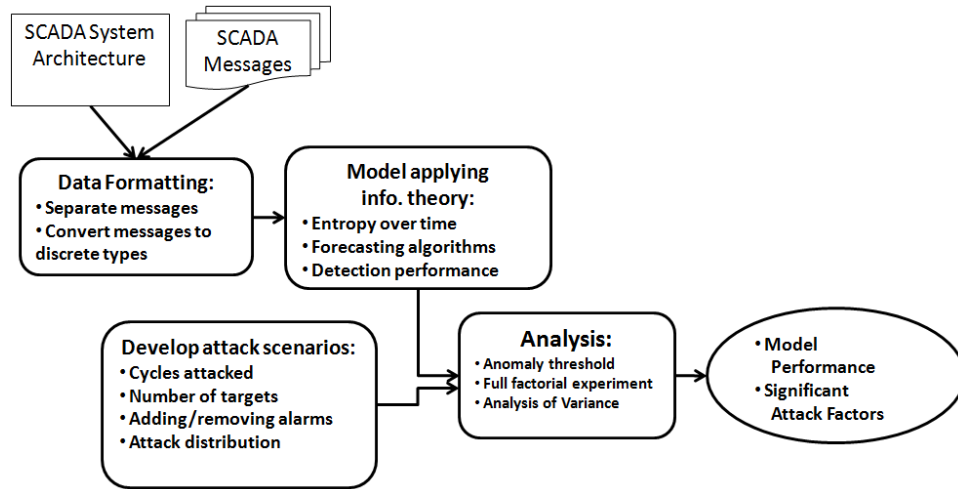
Figure 1. Research framework.

Research Scope and Assumptions

This research focuses on SCADA messages sent from the RTUs to the HMI. Attacker manipulation of the messages is limited to the following:

- Injecting alarms

- Suppressing alarms

This research assumes that the attacker has sufficient access to the SCADA network to be able to create, change, and block messages from the RTU to the HMI. The research also assumes that the person using the model has access to the data to populate it. Examples such as Stuxnet show that an attacker can gain access to a SCADA network to collect and manipulate data.

Thesis Organization

        Chapter II reviews the literature concerning the focus areas of this research. The

focus areas include SCADA vulnerabilities, information theory, and intrusion detection.

Chapter III provides the methodology for applying information theory and conducting the

analysis. This chapter describes the model used in this research. Chapter IV describes

the analysis and results from applying the methodology to a dataset of water treatment

plant SCADA message traffic. Chapter V discusses the conclusions drawn from the

results and potential for future research in this area.

II. Literature Review

The literature review provides an introduction to alarm management and enforces the motivation for studying the vulnerability of message manipulation. The research reviews human factors studies on operator alarm management and reactions to alarms. A key focus area is how operators have the potential to be conditioned to certain responses, such as too many false alarms. The literature review then gives an overview of a typical supervisory control and data acquisition (SCADA) system and discusses vulnerabilities of a cyber attack. The research also reviews current SCADA anomaly detection techniques. The concept of information theory and its use in anomaly detection and other relevant applications are discussed.

Alarm Management

Control systems employ automated alarm management systems, which provide a signal to an operator when an alarm condition is met. These systems are designed to improve operator performance and reduce workload. "The control room indicators and alarms [are] the primary sources of information for monitoring" (Mumaw, Roth, Vicente, & Burns, 2000:41). These systems are not perfect and can miss alarms or report false alarms (Dixon & Wickens, 2006:474). This section reviews studies on operator management of alarms and the characteristics that make them susceptible to attack. The focus area of this research is the vulnerability of an attacker adding alarms and/or hiding alarms.

Alarm Management Environment

SCADA networks "are becoming increasingly large and complex which create special challenges for human operators who must monitor these networks for safe operation" (Su & Yurcik, 2005:1). Nuclear power plants have complicated control rooms that display hundreds of indicators in different ways. Figure 2 shows part of a control room where operators use computer displays, light panels, and other indicators to monitor the plant (Mumaw *et al.*, 2000:40). There are many ways to display information and assist the operator in performing their duties. Guidelines and recommended practices for graphical displays are provided to industries that use SCADA systems (Gerard, 2005:1).

Figure 2. Nuclear power plant control room. (Mumaw *et al.*, 2000:40)

Alarm management involves a balance of providing enough information to an operator to effectively control the system, but avoid overloading them with unnecessary

data. The data is usually presented to the operator with some type of graphical display, which is specific to the industry and the SCADA vendor (Shaw, 2006:145). The display may show the data in real time, use graphs and charts, incorporate a model of the processes, and can be tailored by the operator. Audio or flashing light cues can also bring attention to problems (Mumaw *et al.*, 2000:40). Alarms are often filtered based on priority, operator responsibility, or to block alarms generated when equipment is being repair or replaced (Shaw, 2006:158-159). The operators' heavy reliance on alarms, and subsequent susceptibility to an attacker manipulating alarms, is the motivation for this research.

Human factors studies, such as the research by Wang and Liu (2009), seek to improve SCADA operator effectiveness by changing the human machine interface (HMI). In their research, Wang and Liu study the HMI of a wastewater treatment system. They revealed that the operators were overwhelmed by excessive and unnecessary alarms. "The original system was so designed that it could monitor the system in a very sensitive way and display and [record] all the alarms in detail" (Wang & Liu, 2009:1213). The authors applied solutions such as changing the display of alarms to the operators and filtering alarms based on priority; these changes improved the performance of the operators. This study shows the impact of too many alarms and problem alarms. The next section describes more of the issues operators encounter with managing alarms.

Difficulties in Alarm Monitoring

Mumaw, Roth, Vicente, and Burns published an article in the journal Human Factors (2000) where they studied the alarm management of nuclear power plant operators. In this article they summarized the difficulties of alarm monitoring, "which is

influenced by system complexity and reliability, alarm system design, displays and controls design, and the design of system automation" (Mumaw *et al.*, 2000:44). A complex system could be made up of thousands of elements and "because there are so many interactions among components, subsystems, and instrumentation, it is difficult to derive the full implications of the current failures to determine what state any particular parameter should be in" (Mumaw *et al.*, 2000:44). With a large number of components, there may be continual alarms occurring due to failures or because some part of the system is under repair. The difficulty in interpreting alarms and identifying the state of the system, such as the pipeline accidents, may make operators more susceptible to an attack involving alarm manipulation.

Many situations increase the difficulty in alarm monitoring. Nuisance alarms may distract an operator or decrease their response to an alarm. "For example, multiple alarms can appear for the same event and thereby make interpretation more difficult" (Mumaw *et al.*, 2000:45). In addition, "if a particular parameter is rapidly cycling above and below the alarm set point, an almost continuous stream of [alarms] is generated" (Mumaw *et al.*, 2000:45). Nuisance alarms increase the operator's workload and force them to work harder to determine the important alarms. An attacker could cause these situations to occur and take advantage of the increased workload on the operator.

There are other problem alarms in addition to nuisance alarms. "Flooding alarms" are "alarms caused by routine operations or simple activities." These could be from poorly set thresholds or maintenance on the system. "Stale alarms" are "alarms that [are] unsettled for [a] long time and were overlooked." These may be low priority alarms or alarms from a system problem that is no longer relevant. "Unclear alarms" are from

11

"mis-operation [and are] caused by poor alarm design." Each of these alarms increases

the workload of the operator and adds to the difficulty of determining the state of the

system (Wang & Liu, 2009:1214). An attacker might be able to mimic these types of

alarms and influence the behaviour of an operator. This research looks at an attacker

adding these types of alarms, which could reduce the effectiveness of the operator.

False Alarms

Responding to alarms is a critical task of SCADA operators. Alarms can indicate

significant problems such as pipeline leaks, equipment overloading, and loss of

availability to customers. Due to the importance of real problems, the "system designer

sets the threshold for an alarm so that virtually no true alarm conditions will fail to set off

an alarm. The result is that occasional alarms occur when no true alarm exists (a false

alarm)" (Gerard, 2005:3). Too many false alarms lead to trouble for the operator. A

safety recommendation report by the National Transportation Safety Board (NTSB)

describes the significance of false alarms for pipeline operators:

> If a controller responds to a false leak alarm, the economic cost of shutting down
> the line is small compared with the possibility of spilling a large amount of
> product. However, as the number of false alarms increases, so does the cost of
> responding to all of them. Controllers may try to differentiate false alarms from
> true alarms and respond only to the latter. As a result, they may miss a true alarm,
> increasing the severity of a product leak. (Gerard, 2005:3)

The NTSB report lists alarm management as one of the top five areas for improvement in

pipeline SCADA systems (Gerard, 2005:1).

The continued presence of false alarms can lead to conditioning of the operator.

The NTSB report describes two specific accidents in the pipeline industry resulting from

excessive false alarms. In these accidents, false alarms occurred repeatedly during

certain events. Later, when the alarms were real, the operators assumed they were the same false alarms they had seen in the past. (Gerard, 2005:3) An attacker also may be able to condition an operator to change their behavior by introducing false alarms before a real system problem is generated. The next section continues the discussion of alarms and false alarms.

Reliance and Compliance

Reliance and compliance are two important concepts concerning how operators interact with automated alarm systems. Reliance is a measure of how much an operator trusts the system to detect problems and report alarms. When an operator is reliant on the system, they do not have to spend their time checking that problems are caught; this allows them to spend their time on concurrent activities. Compliance refers to how the operator responds when notified of an alarm. When an operator is compliant, they give immediate attention to every alarm and take appropriate actions. Changes in alarm detection rate and false alarm rate can affect an operator's reliance and compliance of the system. (Dixon & Wickens, 2006:475)

In a study looking at automation reliability in unmanned aerial vehicle (UAV) control, Dixon and Wickens state that "false alarms are well known to cause annoyance, to lead to unnecessary evasive actions, and, in the worst-case scenario, to lead to sufficient distrust of the automated system that true alarms are ignored – the "cry wolf" syndrome" (2006:475). False alarms influence operators' compliance. Too many false alarms and the operator may distrust the system and take longer to respond to an alarm or not respond at all (Dixon & Wickens, 2006:476). This concept is a primary reason that operators are susceptible to having their behaviors affected by message manipulation.

13

Reliance can also be impacted by message manipulation. First, a highly reliable system can lead to an operator fully trusting the automated alarm detection and misses could occur; an alarm event could occur without the alarm detection system reporting it. If an operator is used to working with a reliable system, they are susceptible to the system missing an alarm. As the reliability of the system to catching alarms decreases, the operator involvement increases. If the automated system does not detect some true alarms then the operator must take time away from concurrent tasks to monitor the raw data. (Dixon & Wickens, 2006:475) It may be possible for an attacker to create one of these situations by adding false alarms to the system or hiding true alarms.

Dixon and Wickens conducted a series of experiments on automated alarm systems using humans navigating UAVs with a simulator (2006:476). The experiments evaluated how well the operators performed tasks and monitored for system failures (SFs). Dixon and Wickens (2006) changed the reliability of the automated alarm notification system, which they called *imperfect automation*. The baseline for the experiments was with no alarm aid. They found that "false alarms hurt the system-monitoring task by reducing SF detection rates and increasing SF detection times as compared with baseline" (Dixon & Wickens, 2006:480). Their analysis of the correlations showed "a strong effect of miss rate on reliance (r = .67), as participants became less trusting of the automation to alert them if a failure occurred and...allocated more attention to monitoring the raw data at the expense of two concurrent tasks" (Dixon & Wickens, 2006:484-485). This also had the effect that operators caught more of the misses since they were watching the raw data closer. With a correlation of r = .49, the authors found that increased false alarms decreased compliance, "reflecting the "cry

wolf" phenomenon" (Dixon & Wickens, 2006:485). An important vulnerability

identified by Dixon and Wickens is that "it appears that a false-alarm prone system may

leave the operator somewhat less inclined to pay any attention to the entire automated

domain, whether it be its alerting signal or the raw data contained within" (Dixon &

Wickens, 2006:485). Similarly, operators may be susceptible to changing their behavior

based on an attacker adding false alarms to the system.

SCADA Overview

"[SCADA] systems are used to monitor and remotely control critical industrial

processes, such as gas pipelines, electric power transmission, and potable water

distribution/delivery" (Shaw, 2006:3). Since the 1960s, the basic components of a

SCADA system, shown in Figure 3, are the central operator computer, communications

infrastructure, remote terminal units (RTUs), and plant equipment. The central operator

computer, also known as the SCADA master or host, displays information and receives

commands from the operator via the human machine interface (HMI). Messages and

commands are sent over the communications infrastructure that can consist of one or

several different mediums such as telephone lines, fiber optics, cellular, wireless systems,

and more recently the internet. These messages use either a proprietary or standardized

protocol. The RTUs are the interface between the communications infrastructure and the

plant equipment. Programmable logic controllers (PLCs), which are essentially RTUs

with microprocessors that can perform calculations and programming tasks, may

interface with or replace the RTUs (Shaw, 2006:365). The plant equipment consists of

sensors and control equipment, such as actuators, valves, and switches; these are referred to as slave devices. (Shaw, 2006:3-5)
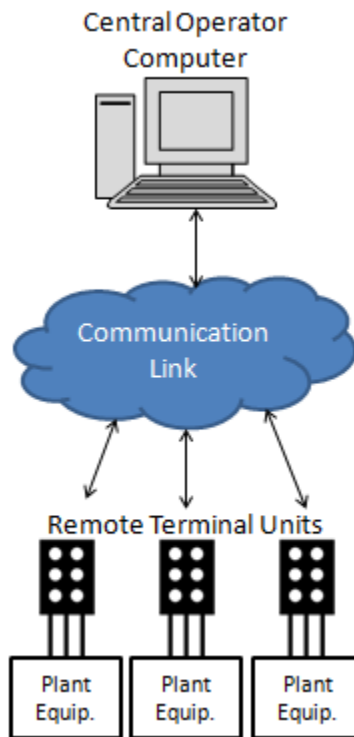
Figure 3. Diagram of a SCADA system. (Shaw, 2006:4)

A SCADA system operates in real time, providing messages and control capability to an operator. The messages consist of equipment/sensor measurements and alarms when something is outside normal operating thresholds. The SCADA master repeatedly polls the RTUs for information on the industrial equipment (Shaw, 2006:5). The operator receives status messages and alarms on displays and/or print outs; he or she then makes decisions on what changes to make or to do nothing. The alarms can be filtered based on the status of the system, such as equipment being replaced or maintained (Shaw, 2006:158).

There are a few different techniques for communication in SCADA systems; this research looks at the polled approach. "The master is in total control of the communication system and makes regular (repetitive) requests for data to be transferred to and from each one of a number of slaves" (Clarke & Reynders, 2004:31). Polling is set up as a full cycle through all the RTUs, one at a time, or as a "high and normal priority arrangement" as shown in Figure 4 (Clarke & Reynders, 2004:33). Full cycle polling was used in this research.
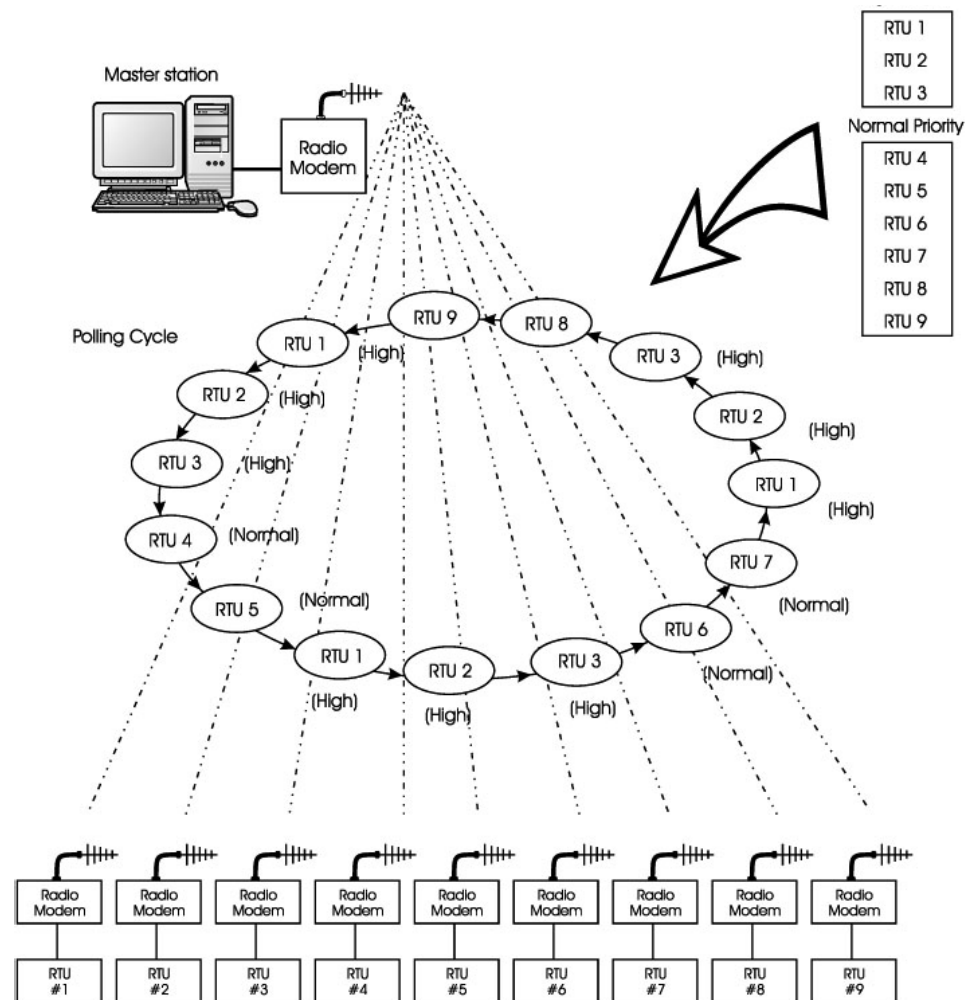
Figure 4. Priority based RTU polling. (Clarke & Reynders, 2004:33)

SCADA Communication Protocols

Most modern SCADA systems communicate using well-developed standard protocols. The proprietary and unique systems of the past were too expensive to maintain and update. Today, large amounts of data are displayed and stored from thousands of sensors located in geographically separated areas. Standardized protocols allow for easier installation and maintenance, and allow a company to purchase products from a variety of vendors. (Clarke & Reynders, 2004:14-15) The methodology for this research can be tailored to many different SCADA protocols, but this research focuses on Modbus.

According to a Control Engineering survey, "Modbus is the most popular industrial protocol being used today...it is simple, inexpensive, universal, and easy to use" (McConahey, 2012). Modbus has been around for over 30 years and is broadly used by SCADA systems owners and equipment makers (Clarke & Reynders, 2004:45). This research is interested in the messages sent from a slave device to the master. Modbus has a simple format for these messages.

The Modbus protocol is framed around the SCADA master communicating with slave devices. The slave device is made up of "four different data types: coils, discrete inputs, input registers, and holding registers" (Clarke & Reynders, 2004:48). Coils are binary circuits which designate if something is ON/OFF or OPEN/CLOSED. The input data types are for receiving commands from the master. The holding registers (registers) contain the value of a reading such as temperature and pressure. The status of coils and values of registers are the data that set off alarms for the SCADA operator, which is the focus of this research.

"A transaction consists of a single request from the host to a specific secondary device and a single response from that device back to the host. Both of these messages are formatted as Modbus message frames…", as shown in Figure 5 (Clarke & Reynders, 2004:47). The bytes are in hexadecimal format for each field.

Figure 5. Modbus message format. (Clarke & Reynders, 2004:47)

The *Address Field* designates which controller (slave device) the message concerns: both outgoing messages from the master and incoming responses from a slave device. The slave device can take on an address identifier between 1 and 247, though most SCADA systems use only a fraction of these. The *Function Field* specifies the function to be performed by the slave device. There are 10 typical functions that perform things such as sending commands, reading the status of coils, and reading registers. The scope of this research includes function codes 1 and 3. Function code 1, *Read Coil Status*, "allows the host to obtain the ON/OFF status of one or more logic coils in the target device" (Clarke & Reynders, 2004:49). The HMI generates an alarm if a coil status is in the wrong position. Function code 3, *Read Holding Register*, "allows the host to obtain the contents of one or more holding registers in the target device" (Clarke & Reynders, 2004:50). The HMI generates an alarm if the value of the register is outside set thresholds. The *Data Field* varies based on the function of the message. Requests from the host contain data about the particular function such as which coils or registers

19

the message concerns. The *Error Check Field* contains the data for error checking the message to assure that "devices do not react to messages that may have been changed during the transmission" (Clarke & Reynders, 2004:47). Chapter III provides more detail on Modbus messages, including formatting the hexadecimal messages in the data log into a more useful form.

SCADA Security

SCADA systems began as isolated systems running proprietary software and custom made hardware. These systems are shifting to standard hardware, software, and communication protocols; they are also incorporating information technology (IT) solutions. A dedicated attacker can learn essentially everything they need to know about a system to perform an attack similar to the ones described in this research due to the increasing use of standardized protocols for SCADA systems, software that often runs on standard Widows or Unix based systems, and the availability of SCADA hardware and software for anyone to purchase (Shaw, 2006:241-242). Using standardized assets and protocols lowers purchase and maintenance costs and increases efficiency, but it "increases the possibility of cyber security vulnerabilities and incidents" (Igure, Laughter, & Williams, 2006:498; Stouffer *et al*., 2008:1).

In the past, SCADA systems were often isolated and may have required physical access. Today many systems have connections to the internet and hackers. The attacker could be a current or former insider, or someone that has gained access to the system. Insider threats are high risk due to their extensive knowledge and access to the system;

they may also be targets by outside threat agents for bribing or stealing their knowledge and access. (Shaw, 2006:239-241)

This research focuses on the vulnerability of an attacker changing data sent to the SCADA master, thereby influencing the actions of a SCADA operator. This is an important vulnerability because "it is primarily the system operators who interact with the system and use the system to monitor and control the target process and field equipment" (Shaw, 2006:137). Due to the way that SCADA systems communicate, it is not difficult for an attacker to emulate message traffic. This risk is compounded because most equipment does not perform checks to verify where the messages originated (Shaw, 2006:60-61). An attack called Man-in-the-Middle can occur, whereby an attacker can intercept and inject messages and/or commands from one part of the system to another. In this attack, the messages appear to be from a legitimate source, though an attacker has the control. "The attacker may be able to cause invalid data to be displayed on a console or create invalid commands or alarm messages" (U.S. Department of Homeland Security, 2011:33).

SCADA system owners (or an attacker) have the capability to collect the message traffic for the model defined in this research by using a packet sniffer. A packet sniffer is a device or program that can collect and save message traffic going through a specific place in the network. One use of a packet sniffer is analyzing message traffic, such as collecting statistics on message types and frequencies. (Jung, Song, & Kim, 2008:78-79) Legitimate commercial companies develop packet sniffers for network security. Chapter III also describes the capabilities of commercially available software to log messages.

Intrusion Detection

Dorothy E. Denning described a model for intrusion detection in her highly cited 1986 paper *An Intrusion-Detection Model* (Denning, 1986). Denning's "model is based on the hypothesis that exploitation of a system's vulnerabilities involves abnormal use of the system; therefore, security violations could be detected from abnormal patterns of system usage" (Denning, 1987:222). She also discussed how "security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage" (Denning, 1987:222). This research uses this concept of intrusion detection.

Intrusion detection systems (IDS) fall into two categories: signature detection and anomaly detection. Signature detection works by scanning the network and looking for characteristics matching previous intrusions of the system. With properly updated libraries of signatures, signature detection can achieve high detection rates and low false alarm rates. Attackers can thwart signature detection by using new techniques or new variants of existing techniques. Anomaly detection does not look for signatures but instead watches for deviations from normal operations. It uses statistics gathered from the usual behavior of the system and identifies when behavior is outside certain thresholds. Anomaly detection has the possibility of catching new attacks, but its performance is a balance of increasing the sensitivity to false alarms and decreasing missed detections. (Zhu & Sastry, 2010:82) The model in this research, which is described in Chapter III and analyzed in Chapter IV, used anomaly detection and not signature detection.

Denning provides a methodology for anomaly detection using a mean and standard deviation statistical model. This model defines an event as abnormal if it falls a

predefined number of standard deviations outside the mean of the baseline state. If d is the number of standard deviations, Chebyshev's inequality states that "the probability of a value falling outside this interval is at most $1/d^2$; for $d = 4$, for example, it is at most 0.0625" (Denning, 1987:225).

Zhu and Sastry summarize the issue of security on SCADA systems by stating: "SCADA systems were designed without cyber security in mind and hence the problem of how to modify conventional Information Technology (IT) intrusion detection techniques to suit the needs of SCADA is a big challenge" (Zhu & Sastry, 2010:77). The authors evaluate several IDS's for use on SCADA systems with varying results. The authors concluded that one of the best IDS's evaluated involved using the Modbus/TCP protocol. The Modbus protocol is "the most widely used application layer protocol for communication between control station to field devices in industrial networks" (Zhu & Sastry, 2010:83). The IDS evaluated by Zhu and Sastry was a signature-based IDS. Though the research of this thesis involved anomaly detection instead of signature-based detection, the author's reasons for using Modbus still apply.

There are other examples of anomaly-based detection on SCADA systems. Yang, Usynin, and Hines (2006) used a form of anomaly detection called pattern matching. It uses normal system traffic to "build a traffic and usage profile for a given network...[and] when new traffic data fails to fit within a predetermined confidence interval of the stored profiles, then an alarm is triggered" (Yang, Usynin, and Hines, 2006:13). The authors state that this kind of model is "is based on the hypothesis that security violations should change the system usage and these changes could be detected" (Yang *et al.*, 2006:13).

Information Theory

"Information is the resolution of uncertainty."

- Claude Shannon

Information theory, as developed by Claude Shannon, describes the "fundamental laws of data compression and transmission" and is a "unifying theory with profound intersections with Probability, Statistics, Computer Science, and other fields..." (Verdu, 1998:2057). This section outlines many of the key concepts of information theory and describes some of its applications outside the fields of compression and transmission. The emphasis is on applications relevant to anomaly detection. Areas where the use of information theory has failed are also discussed.

Shannon's Entropy

In 1948, C. E. Shannon wrote a paper in <u>The Bell System Technical Journal</u> titled *A Mathematical Theory of Communication*, which outlined a "measure of the information produced" from a set of messages out of a finite set of possible messages (Shannon, 1948:379). This section outlines the information theory concepts and metrics developed by Shannon that are relevant to this research. These include bits (binary digits) as a unit of measure and the entropy of a message source. The application of these concepts involves a finite number of discrete messages.

Shannon defined a general communication system as consisting of five pieces: information source, transmitter, channel, receiver, and destination. The information source is the origin of the information that goes into the message. The transmitter transforms the message into a format for sending over the channel. The channel is the medium for sending the message. The receiver converts the signal sent over the channel

back into a readable message. The destination is the intended recipient of the

information. Figure 6 shows a diagram of this communication system. (Shannon,

1948:380-381) Noise can occur at any point in the transmission and may cause the

message at the destination to be different than the message sent from the information

source (Shannon, 1948:398).
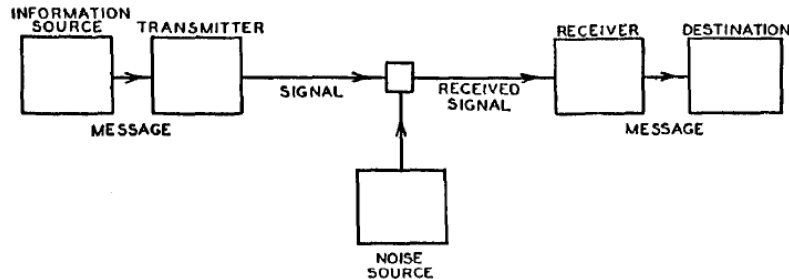
Figure 6. Shannon's general communication system diagram. (Shannon, 1948:380)

Information Entropy

In the book <u>Elements of Information Theory</u>, Cover and Thomas define entropy

as "a measure of the uncertainty of a random variable"; this measure involves "a discrete

random variable with alphabet χ and probability mass function $p(x)$" (Cover & Thomas,

2006:13). Here, χ is the set of possible messages sent from the information source to the

destination. Entropy describes how much information is gained from some source. If a

source always sends the same message then no information is gained and the entropy is

equal to zero. If a source is equally likely to send any of its possible messages, then the

amount of information is high and depends on the number of possible messages; this is

maximum entropy.

The simplest situation is looking at the maximum entropy, which is the total

amount of information possible from an information source. This depends on the number

of possible message types and occurs when the message possibilities are all equally

likely. The entropy, H, for a source with n equally probable message types (Shannon, 1948:379-380) is

$$H = \log_2 n \qquad (1)$$

where the log is in base 2 for units of bits (for the rest of this document, any log is implied to be base 2). It is assumed that $0 \log 0 = 0$. For example, a fair coin has two $(n = 2)$ equally likely outcomes: heads and tails. The entropy of a single fair coin is: $\log 2 = 1 \; bit$.

In most applications, including this research, each message is not equally likely. In this case, the entropy is the sum of the information from each individual message, weighted by its probability. Using the random variable X for the message type, the entropy is

$$H = -\sum_{x \in X} p(x) \log p(x) \qquad (2)$$

where $p(x)$ is the probability that X equals x (Cover & Thomas, 2006:14). "Note that the entropy...does not depend on the actual values taken by the random variable X, but only on the probabilities" (Cover & Thomas, 2006:14). In other words, the information from a communication source depends on the likelihood of each message. Here, the negative symbol comes from changing $\frac{1}{n}$ to $p(x)$: $\log n = -\log n^{-1} = -\log \frac{1}{n} \rightarrow -\log p(x)$ (Cover & Thomas, 2006:14). Another way to write entropy is by using the expectation operator, E, for some function $g(x)$: $E[g(x)] = \sum_{x \in \chi} p(x)g(x)$ (Cover & Thomas, 2006:14). Entropy (Cover & Thomas, 2006:14) now becomes

$$H = -E[\log p(X)] \qquad (3)$$

Using the coin example, an unfair coin with a 0.25 probability of heads and a 0.75 probability of tails is: $H_{coin} = -0.25 \log 0.25 - 0.75 \log 0.75 = 0.81 \, bits$. This is less entropy than the fair coin, which had 1 bit. Another way to describe entropy is the weighted sum of the information for each possible outcome. Using the unfair coin example, most of the time the coin flip outcome is going to be tails. Since this is expected, less information is gained. Some of the time the coin will show a heads; this is less expected and more information is gained. The entropy of the coin weights both outcomes on their probability of occurring. The fact that entropy is independent of the content of the messages is key to developing the methodology of this research. The content of specific messages could be complicated, but using only the frequencies of the messages allows the information to be quantified and direct comparisons between states of the system can be made. In a SCADA system, one can calculate the entropy of the system based on the number of messages that are alarms. As discussed later, several researchers, such as Lee and Xiang (2001), have used entropy for anomaly detection.

Joint and Conditional Entropy

When two discrete random variables are of interest then it is useful to calculate the joint entropy and conditional entropy. For example, consider two random variables X and Y, where X is the number of alarms in a SCADA system over some period of time and Y is the number of alarms out of the last 10 messages. For these random variables X and Y, with joint distribution $p(x, y)$, the joint entropy (Cover & Thomas, 2006:16-17) is

$$H(X, Y) = -E[\log p(X, Y)] \tag{4}$$

This is another form of calculating the entropy of a source, except now two outcomes are of interest. Here, joint entropy uses the probability of $X = x$ and $Y = y$. In the SCADA

example, $H(X, Y)$ is the joint entropy of the total number of alarms and the number of alarms in the last 10 messages.

The conditional entropy of two random variables (Cover & Thomas, 2006:17) with the probability of Y given X of $p(Y|X)$ is

$$H(Y|X) = -E[\log p(Y|X)] \qquad (5)$$

where the expectation operator sums over all values of the joint distribution. Again using the SCADA example, it may be useful to know the entropy for the alarms in the system based on knowing the outcome of the last 10 alarms. The chain rule,

$$H(X, Y) = H(X) + H(Y|X) \qquad (6)$$

is a useful theorem which states the joint entropy in terms of the entropy of one random variable and the conditional entropy (Cover & Thomas, 2006:17). Discussed later, Arackaparambil, Bratus, Brody, and Shubina (2010) used conditional entropy for anomaly detection in network traffic.

Relative Entropy and Mutual Information

Two other information-theoretic concepts involving more than one random variable are relative entropy and mutual information. Relative entropy describes the distance between two distributions, represented by $D(p||q)$. It "is a measure of the inefficiency of assuming that the distribution is q when the true distribution is p" (Cover & Thomas, 2006:19). For example, if the wrong distribution is used, then the information source is described by the entropy of the wrong distribution plus the relative entropy (Cover & Thomas, 2006:19). Relative entropy (Cover & Thomas, 2006:54) is

$$D(p||q) = \sum_{x \in \chi} p(x) \log \frac{p(x)}{q(x)}$$

$$= E\left[\log \frac{p(X)}{q(X)}\right]$$

(7)

A SCADA example of using relative entropy is comparing the system from one day to the next. Large changes in the two day's distributions may indicate an anomaly has occurred. Relative entropy "is known under a variety of names, including the Kullback-Leibler distance, cross entropy, information divergence, and information for discrimination" (Cover & Thomas, 2006:54).

Conditional relative entropy can also be calculated, which is the same as relative entropy except both distributions are now conditioned on some event. It is "the average of the relative entropies between the conditional probability mass functions $p(y|x)$ and $q(y|x)$ averaged over the probability mass function $p(x)$" (Cover & Thomas, 2006:24), which is

$$D(p(y|x)||q(y|x)) = \sum_{x \in \chi} p(x) \sum_{y \in \chi} p(y|x) \log \frac{p(y|x)}{q(y|x)}$$

$$= E_{p(x,y)}\left[\log \frac{p(Y|X)}{q(Y|X)}\right]$$

(8)

A SCADA example of relative conditional entropy is comparing two days' alarm message distributions based on the number of alarms from the previous day. This may help detect how much the system is changing and help determine if an anomaly has occurred.

The mutual information, $I(X;Y)$ (Cover & Thomas, 2006:21), can be written in terms of marginal and joint entropies, and marginal and conditional entropies:

29

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

$$= H(X) - H(X|Y) \qquad (9)$$

$$= H(Y) - H(Y|X)$$

It is "the reduction in the uncertainty of X due to the knowledge of Y" (Cover & Thomas, 2006:21). Mutual information measures how much information is gained for some event, if some other event is already known. For example, mutual information can describe how much the entropy of a SCADA system changes based on knowing what occurred the previous day. Mutual information was used by Xia, Qu, Hariri, and Yousif (2005) for classifying data, as discussed later.

Entropy Error Estimation

Estimating the errors in calculated entropies is useful, especially when dealing with smaller sample sizes. Mark Roulston published a paper in *Physica D* where he derived and evaluated formulas for entropy errors based on the works of Basharin (1959), Harris (1975), and Herzel and Grosse (1997). The error in the entropy is calculated from observed entropies, H_{obs}, from a set of data containing N samples, with B different states, where each sample is in state i ($i = 1,2,...B$). The predicted range of the true entropy of the system is

$$H_{true} \approx H_{obs} + \frac{B^* - 1}{2N} \pm \sigma_H,$$

$$\sigma_H = \sqrt{\frac{1}{N} \sum_{i=1}^{B} (\log q_i + H_{obs})^2 \, q_i(1 - q_i)} \qquad (10)$$

where B^* is the number of states with non-zero probabilities and q_i is the observed

distribution for state i; σ_H is the standard error of the observed entropy (Roulston,

1999:285-286,293).

The error for mutual information is

$$I_{true} \approx I_{obs} + \frac{B_X^* + B_Y^* - B_{XY}^* - 1}{2N} \pm \sigma_I,$$

$$\sigma_I = \sqrt{\frac{1}{N}\sum_{k=1}^{B_X}\sum_{l=1}^{B_Y}(\log q_k^X + \log q_l^Y - \log q_{kl} + I_{obs})^2 q_i(1 - q_{kl})}$$

(11)

for distributions X and Y, where $q_k^X = \sum_{j=1}^{B_Y} q_{kj}$ and $q_l^Y = \sum_{i=1}^{B_X} q_{il}$ (Roulston, 1999:293).

These measures quantify the error in determining the true system entropy and mutual

information, based on the variance of observed messages. These expressions may give

insight into what a system owner considers an allowable range of the entropy of the

system. If the range is large, the system may be vulnerable to missing attacks that do not

exceed the set entropy range.

Information Theory for Anomaly Detection

The use of information theory for anomaly detection is outlined in the work by

Lee and Xiang, where they explored the use of "several information-theoretic measures,

namely, entropy, conditional entropy, relative conditional entropy, information gain, and

information cost for anomaly detection" (Lee & Xiang, 2001:130). They define how

these measures can be used for anomaly detection and then apply them to different

datasets. Each of these, except for information cost, is discussed in this section. Lee and

Xiang define information cost as "the average time for processing an audit record and

31

checking against the detection model" (Lee & Xiang, 2001:133). This concerns the speed of the model when real-time analysis is needed and is outside the scope of this research.

The first information theory anomaly detection method described by Lee and Xiang is entropy, as defined in equation (2). Entropy can be used "as a measure of the regularity of the audit data" by representing "*unique* records" as a "class". The authors continue by stating that entropy as an anomaly detection tool is best used for data with smaller entropy due to higher regularity; an "anomaly detection model constructed using dataset with smaller entropy will likely be simpler and have better detection performance." (Lee & Xiang, 2001:131) This method may be successful for a SCADA system using Modbus due to the periodic polling of the slave devices. The SCADA messages can be put into classes of alarm or not an alarm.

Another method is using conditional entropy, as defined in equation (5). Conditional entropy describes the amount of uncertainty remaining when an event has been observed, such as a subset of the audit events. The authors state that conditional entropy is useful "because of the temporal nature of user, program, and network activities, we need to measure the temporal or sequential characteristic of audit data" (Lee & Xiang, 2001:131). Lee and Xiang state that conditional entropy is useful "as a measure of regularity of sequential dependencies...[and] the smaller the conditional entropy, the better" (Lee & Xiang, 2001:132). Systems with higher conditional entropy are harder to model. The usefulness of conditional entropy for SCADA systems depends on the regularity of alarms over time.

Lee and Xiang discuss how relative entropy, defined in equation (7), can be used to compare two distributions from the same event list. They provide an example comparing test data to training data. "Relative entropy measures the *distance* of the regularities between two datasets" (Lee & Xiang, 2001:132). When conditional entropy is used on the dataset, relative conditional entropy, defined in equation (8), calculates the distance between two datasets. The authors state that smaller entropy is better for both of these methods. Relative entropy may give insight into the impact of message manipulation in a SCADA system. Anomalies may be detected if the entropy of the system is compared day by day (or other time period), based on the entropy of the previous day. It is a way to possibly detect changes in the system due to an attack (or other anomalous event).

Information gain is also discussed as an information theory method for anomaly detection by Lee and Xiang. This is similar to mutual information, as defined in equation (9). Information gain is "the reduction of entropy when the dataset is partitioned according to the feature values" and can be used when "the records are defined by a set of features and each record belongs to a class." Information gain of an attribute A of dataset X is

$$Gain(X, A) = H(X) - \sum_{v \in Values(A)} \frac{|X_v|}{|X|} H(X_v) \qquad (12)$$

where "$Values(A)$ is the set of possible values of A and X_v is the subset of X where A has value v" (Lee & Xiang, 2001:132). Information gain could be used for a system if the message sequence possibilities are grouped into classes. The next section describes examples of using information theory to classify data for anomaly detection.

Classifying Data

Some intrusion detection methods use information theory as a tool to classify data. Often, there is too much data to process in real time and a technique is required to identify when further analysis is needed. This is also true for SCADA communications. Xia, Qu, Hariri, and Yousif published an anomaly intrusion detection system (IDS) for a network, which used "information theory to filter the traffic data and thus reduce the complexity...[and] identify the most relevant features" (Xia, Qu, Hariri, & Yousif, 2005:11, 12). Specifically, they used mutual information between two random variables. Two of the reasons they chose this method were because mutual information "measures general statistical dependence between variables" and it "is invariant to monotonic transformations performed on the variables" (Xia *et al.*, 2005:12). The features included things such as *protocol_type*, *service*, and *logged_in*. They applied mutual information, equation (9), to calculate the amount the uncertainty in the normal/abnormal decision variable was reduced when each feature was used. Xia *et al.* chose the features with the largest amount of mutual information, allowing them to narrow the features of interest from 41 to four. This reduced the amount of data analyzed by their IDS and allowed it to run in real time. The application of information theory also improved the detection rate and lowered the false alarm rate (Xia *et al.*, 2005:16).

Wang, Zhang, Guo, and Li used entropy as a "classification method which can divide Internet traffic into different content types (including Text, Picture, Audio, Video,...)" in real-time for network management (Wang, Zhang, Guo, & Li, 2011:45, 51). Their work showed that traffic can be broken down into finite elements. They looked at each byte in a file and computed the entropy of the entire file. For their analysis, they

used the "standardized Shannon entropy, defined as $H/\log m$, where the normalized factor is the logarithm of [m = total number of items]" (Wang et al., 2011:46). Wang *et al.* went a step further by grouping consecutive bytes into an arbitrary group size k, and found "the entropy of the given file over all possible k consecutive bytes" (Wang et al., 2011:46). The authors describe a simple example of this method:

> ... a file is drawn from a set of n (n = 3) different items {a,b,c}. Let the file M (M = < a,a,a,a,b,b,c >) be the target file under analysis. For instance, we can treat every three continuous bytes as an element and the new sequence of M is < aaa,aaa,aab,abb,bbc >. In our example, the total number of items m = 2 + 1 + 1 + 1 = 5, and the entropy H(X) = −(2/5)log(2/5) − 3 × (1/5)log(1/5) = 1.922. It often would be turned into standardized Shannon entropy, and the final result of standardized entropy is 1.922/log5 = 0.828. (Wang et al., 2011:46)

The methodology of Wang *et al.* was to apply their model to a training set of data and use those results on test data. They improved the speed of the model by using a "partial file space" instead of the entire file. The work of Wang *et al.* shows an example of partitioning a large amount of data into finite groups for calculating entropy. Their approach using partial files may be useful when computing speed is an issue or in cases of extreme amounts of traffic.

Entropy for Anomaly Detection

Shannon's information entropy has been used for intrusion detection in a SCADA system. Benjamin Kahler wrote a paper concerning SCADA system intrusion detection using graph theory (Kahler, 2013). In this paper he describes a graph-based IDS that incorporates information entropy. Kahler's approach builds an adjacency matrix for nodes in a SCADA system from message traffic. A baseline adjacency matrix is created by collecting data from normal operations. The entropy equation, (2), is then applied to calculate the entropy for each node in the adjacency matrix (Kahler, 2013:284). The

network is then monitored for new node connections and any changes in entropy are compared to a threshold of the baseline entropy. "If the new entropy value extends the threshold, an alarm is triggered" (Kahler, 2013:285). Kahler tested his approach using a virtual network of 15 machines and used an internal port scan attack on one of those machines. Figure 7 shows a graph of the entropy for the nodes in the network; it shows the baseline (*standard*) entropy and the entropy for the attack scenario (*case 1*). The outlier due to the network attack on ws-2 is apparent. (Kahler, 2013:285)

	deb1	deb2	xp1	xp2	ws1	ws2	ws3	ws4	ws5	ws6	ws7	ws8	ws9	ws10	ws11
Entropy (standard)	6,15	6,62	4,25	4,02	4,25	6,98	5,32	6,71	5,32	6,71	5,32	5,32	5,32	2,71	5,32
Entropy (case 1)	6,15	6,01	5,32	4,02	5,32	1,30	6,15	6,98	5,32	6,71	5,32	6,15	5,32	2,71	6,15
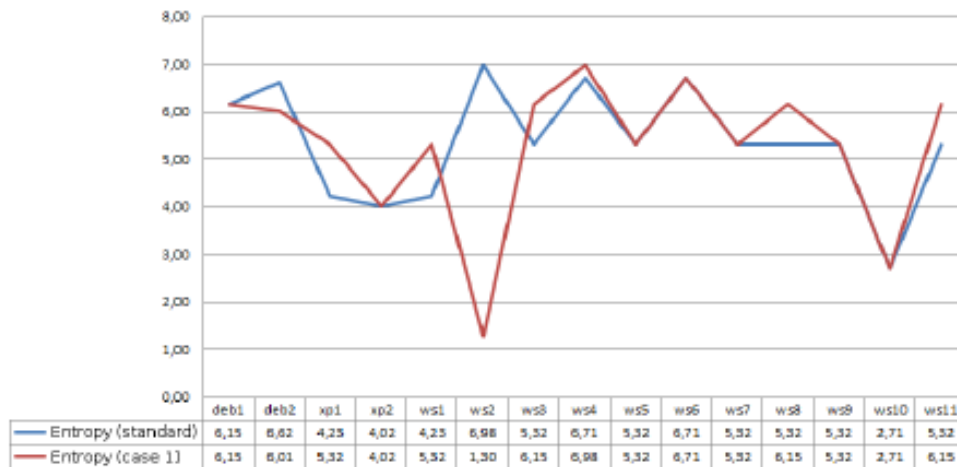
Figure 7. Entropy results from the Kahler paper. (Kahler, 2013:285)

Building on the work of Lakhina, Crovella, and Diot (2005) and Wagner and Plattner (2005), Winter, Lampesberger, Zeilinger, and Hermann proposed "network entropy time series...to reduce high-dimensional network traffic to a single metric describing the dispersion or "chaos" inherent to network traffic" (Winter, Lampesberger, Zeilinger, & Hermann, 2011:194). Their detection algorithm looks for an "abrupt change" in network flows, which is "an unexpectedly high difference between two measurement intervals" that exceeds a defined threshold (Winter *et al*., 2011:194). Winter *et al*. created a detection algorithm focusing on network flows (data on internet

protocol addresses and protocol numbers) because it involves less data than the message content, is easily collected, and can be processed faster.

Wagner and Plattner discussed the tradeoffs of interval length. They stated that "short intervals give fast observation, but [have] sensitivity to short-term effects...longer intervals smoothen out the resulting graphs, but cause a longer reporting latency" (Wagner & Plattner, 2005:174). The tradeoffs are balanced using a commonly used technique called a sliding widow. Here, a window width of time is shifted and overlapped with the previous window.

Winter *et al*. calculated the entropy of five flow attributes using equation (7) and a sliding window approach. "In order to make the result of the entropy analysis easier to interpret, [the authors] normalize it to the interval [0, 1] by using the normalized entropy" (Winter *et al*., 2011:196):

$$H_0 = \frac{H}{H_{max}} = \frac{H}{\log n} \tag{13}$$

Using the sliding window approach, they calculated the entropy over a five minute time period, with four minutes overlapping the previous time period. A "simple exponential smoothing [(SES)]...algorithm is used to smooth time series as well as to conduct short-term predictions" (Winter *et al*., 2011:197). The SES is defined in detail in Chapter III.

> The basic idea for detecting abrupt changes is to continuously conduct short-term predictions and determine the difference between the prediction and the actual measurement. The higher the difference, the more unexpected and hence abrupt the change is. (Winter *et al*., 2011:197)

The algorithm used by the authors does not take into account "trends" or "seasonal components". This choice was made due to the short measurement interval (one minute)

and a cycle of only one day. (Winter *et al*., 2011:197) The authors used a single anomaly

score for their anomaly detection system. The anomaly detection score is a weighted sum

of the errors from the predicted versus actual entropy values, for all five flow attributes.

A threshold for anomalies is set and an alarm is raised if the threshold is busted. (Winter

et al., 2011:198-199)

An example of the results from Winter *et al*. is shown in Figure 8. It shows a day

of data from a university network with injected anomalies. The shaded areas A and B are

events which triggered the threshold alarm using the collected data alone. C and D are

injected anomalies which also triggered an alarm. (Winter *et al*., 2011:202)
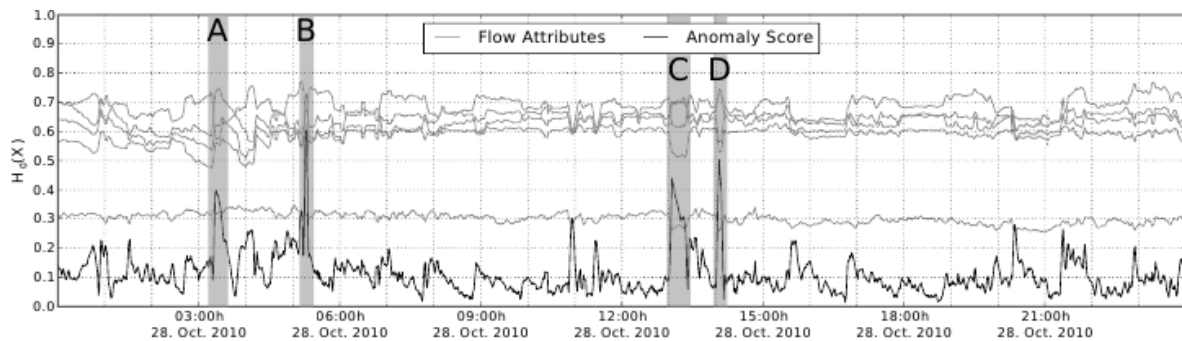
Figure 8. Entropy time series and anomaly score from university data. (Winter *et al*., 2011:202)

The sliding window approach has potential vulnerabilities. A small scale attack

could be used to avoid breaching the anomaly threshold. Winter *et al*. stated that the

detection algorithm could also be overcome if an attacker could "launch an attack in a

slow but continuously increasing way to "stay under the radar" of [the] algorithm"

(Winter *et al*., 2011:203). Attacks that go undetected could cause problems to a SCADA

system such as blocking alarms sent to the HMI or adding false alarms that cause the

operator to distrust the alarm detection system. The authors stated that slow developing

large-scale attacks could be counteracted by adjusting the time scale to a larger range. This type of attack was evaluated during this research in Chapter IV.

Network Traffic Anomaly Detection

Nychis, Sekar, Andersen, Kim and Zhang analyzed different techniques which used information theory measures for anomaly detection of computer network traffic. They stated: "Entropy-based approaches for anomaly detection are appealing since they provide more fine-grained insights than traditional traffic volume analysis" (Nychis, Sekar, Andersen, Kim, & Zhang, 2008:151). Similar to Winter *et al.*, the authors used a dataset of university network traffic and calculated the entropy for different features. To directly compare the entropies using different network features, Nychis *et al.* normalized the entropies using equation (13). Figure 9 shows an example of their analysis; the entropy for five minute "epochs" is plotted over time for different network features (Nychis *et al.*, 2008:153). The graph also shows the traffic volume and the letters at the top indicate different anomalous events, some were detected and others were missed.

Figure 9. Entropy plotted against time for different network features from university network traffic data. (Nychis *et al.*, 2008:153)
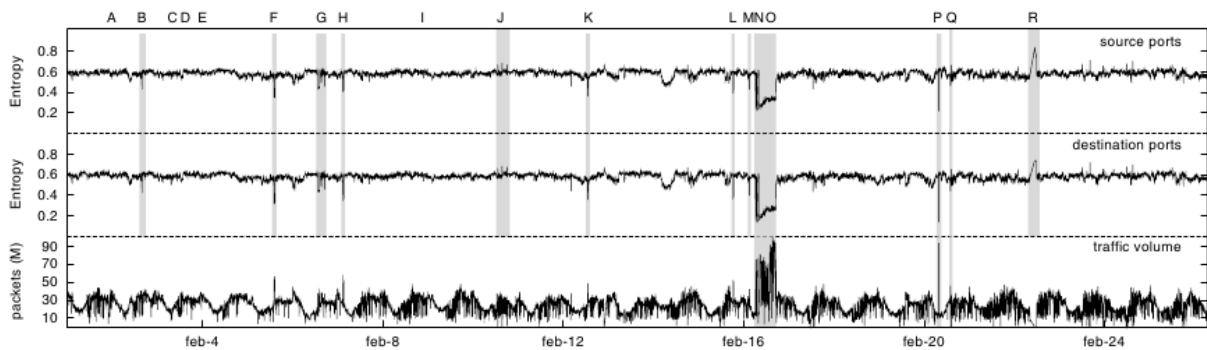
The work of Nychis *et al.* discussed lessons learned for performing traffic anomaly detection using entropy: "select traffic distributions that complement one

another and provide different views into the underlying traffic structure" (2008:155) and "using time-series anomaly detection on the correlation scores can expose new anomalies that do not manifest in the raw time-series" (2008:152). These results have the possibility of being generalized to other anomaly detection applications.

Arackaparambil, Bratus, Brody, and Shubina use an algorithm which calculates conditional entropy for anomaly detection in network traffic. They use entropy because it is a "statistic that measures the variability of the feature under consideration...[and] anomalous activity in network traffic can be captured by detecting changes in this variability" (Arackaparambil, Bratus, Brody, & Shubina, 2010:1). The authors make the case for conditional entropy because it is not easy to determine a baseline "normal" for a system and an attacker could try to imitate the current system profile and slowly make changes to avoid detection by an IDS. Conditional entropy makes it more difficult for an attacker to mask their actions because "maintaining dependencies between features while at the same time carrying out an attack is harder than just maintaining the distribution of features independently" (Arackaparambil *et al*., 2010:2).

Arackaparambil *et al*. applied conditional entropy to a dataset of 802.11 wireless link layer headers collected at Dartmouth College. The features used for conditioning were the source MAC address, frame length, and duration/ID. Since they were using different distributions to calculate conditional entropy, the authors normalized the entropy values "in order to allow for comparisons of these values between different pairs of features" (Arackaparambil *et al*., 2010:4).

One of the methods the authors used for anomaly detection was plotting entropy deviations from baseline over time. Figure 10 shows the authors' application of

conditional entropy; here X is the frame length and Y is an address field (Arackaparambil *et al*., 2010:7). The large deviation in entropy at around 18:30 is from an attack on the network by the authors. Arackaparambil *et al*. demonstrated anomaly detection using conditional entropy.

(a) $H(X|Y)$
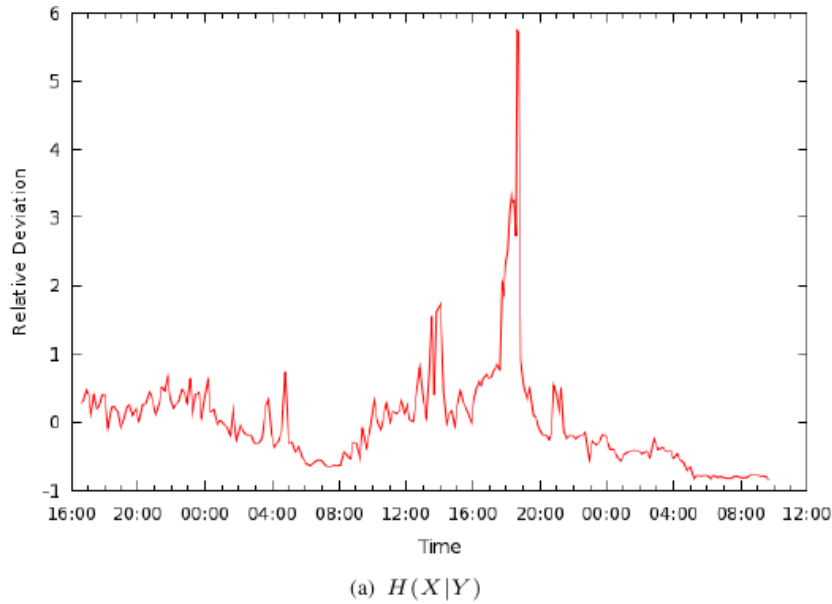
Figure 10. Using conditional entropy for anomaly detection. (Arackaparambil *et al*., 2010:7)

Leak Detection

Leak detection in a pipeline is a form of anomaly detection. Information entropy was used by Zhang, Qin, Wang, and Liang for leak detection in a SCADA-run pipeline system. A concept derived from information theory improved leak detection sensitivity and lowered false alarm rates. (Zhang, Qin, Wang, & Liang, 2009:981) The previous leak detection system was prone to false alarms during normal operations. The authors used the information entropy algorithm (Zhang *et al*., 2009:985)

41

$$H(P, F) = -\sum_{i=1}^{2} p(x_i) \log p(x_i) \tag{14}$$

to combine the information from the pressure and flow rate, where x_1 is pressure (P), and

x_2 is flow rate (F). When the entropy exceeded a certain value, it triggered further

analysis for leak detection. The authors state the usefulness of applying information

theory to leak detection:

> One key feature of the leak detection system is that it has learning capability. In order to optimize the performance of the leak detection system, parameter tuning is carried out both during the design stage and after the initial installation. On the basis of this information fusion system, [leaks] can be [detected] with low false alarm rate and high sensitivity. (Zhang *et al.*, 2009:985)

Failures of Information Theory

Shannon wrote a short article titled *The Bandwagon*, where he warned others of

the overuse of information theory. He stated that information theory has "perhaps been

ballooned to an importance beyond its actual accomplishments" and cautioned that it

"will be all too easy for our somewhat artificial prosperity to collapse overnight when it

is realized that the use of a few exciting words like *information, entropy, redundancy*, do

not solve all our problems" (Shannon, 1956:3). This is true for some attempted

applications of information theory such as psychology.

R. Duncan Luce authored an article in the journal <u>Review of General Psychology</u>

where he describes the "incompatibility between information theory and the

psychological phenomena to which it has been applied" (Luce, 2003:183). For a time

following Shannon's paper, psychologists attempted to apply information theory in their

experiments with little success (Luce, 2003:184-185). Luce states the biggest reason for

this incompatibility is the dependence on structure of signals in much of psychology. In psychology, the response to stimuli has a dependence on "differences or ratios of intensity and frequency measures between pairs of stimuli" (Luce, 2003:185). This is an example where information theory was used because of its popularity and not because it made sense as a technique. However, there is evidence that information theory has been successfully applied to anomaly detection for SCADA systems.

Attack Scenarios

The way an attacker implements the message changes can have an impact on if anomalies are caught by an operator or anomaly detection system. Dorothy Denning described this type of attack in her paper on intrusion detection. She stated that "it may be possible for a person to escape detection through gradual modifications of behavior or through subtle forms of intrusion that use low-level features of the target system" (Denning, 1987:232). Similarly, in a paper about anomaly detection using conditional entropy, Arackaparambil *et al.* state that there is "the concern that an adversary could attempt to mask the effect of his attacks on variability by a mimicry attack disguising his traffic to mimic the distribution of normal traffic in the network, thus avoiding detection by an entropy monitoring sensor" (Arackaparambil *et al.*, 2010:1). SCADA system operators are susceptible to subtle attacks that are too slow to cause an alarm from an intrusion detection system and an operator may not detect the change over time. These types of attacks are of particular interest to this research. Chapters III and IV describe the specific attack scenarios used in this research.

Summary

All over the world, many critical infrastructures rely on SCADA systems. Alarm monitoring is an important part of operating these systems and recent standardization of protocols and internet connections make operators susceptible to an attacker making changes to the system. Though it has been 65 years since Claude Shannon introduced information-theoretic measures for use in telecommunications, the concepts he developed and the work of others in this area have been applied to many fields in science in technology. This chapter reviewed some of the key concepts of information theory and its application to anomaly detection. Chapter III describes the methodology for this research.

III. Methodology

Introduction

The objective of this research is to evaluate the use of information theory in an anomaly detection model and the significance of different attack scenarios. This research proposes to meet this goal by using the measure of entropy derived from information theory applications to anomaly detection (Lee & Xiang, 2001). The desired outcome is an information entropy based model for detecting system problems and message manipulation attacks. The model is used to analyze different attack scenarios using a full factorial experiment repeated for three different model settings. The framework for the research is shown in Figure 11 and summarized in the following paragraphs.
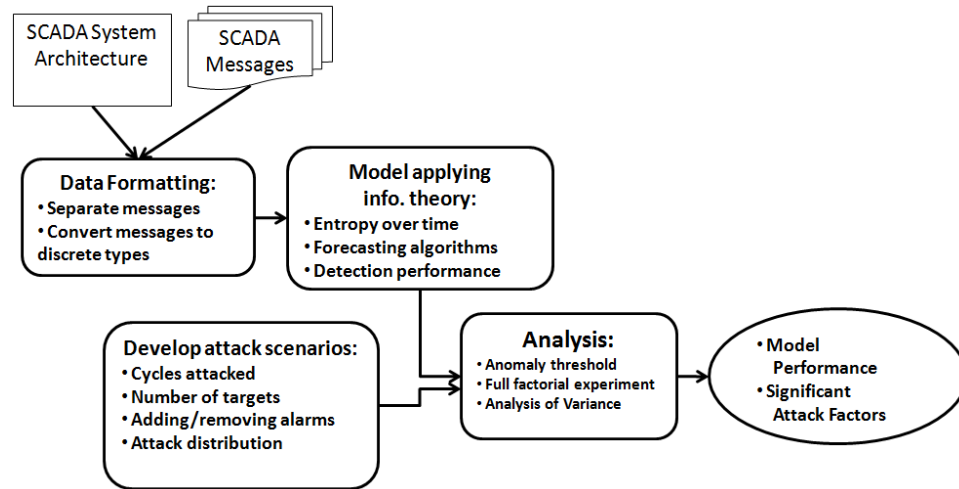

Figure 11. Research framework.

The methodology for this research begins with extracting data from messages collected from a Supervisory Control and Data Acquisition (SCADA) system; specifically, a system using the Modbus communication protocol. The scope of this research includes message traffic data from slave devices to the SCADA master. This

research concerns the messages that hold alarm information, particularly the status of coils and readings of registers. These messages use the Modbus function codes of 1 and 3, which correspond to *Read Coil Status* and *Read Register*, respectively (Clarke & Reynders, 2004:49). Chapter II discusses the function codes in more detail. These two message types provide the data needed for the SCADA master to determine if an alarm has occurred, such as a coil being off when it is supposed to be on or a register value outside of the preset threshold. This methodology assumes that the user has collected the message traffic, imported the data into Microsoft Excel, and converted it to decimal format.
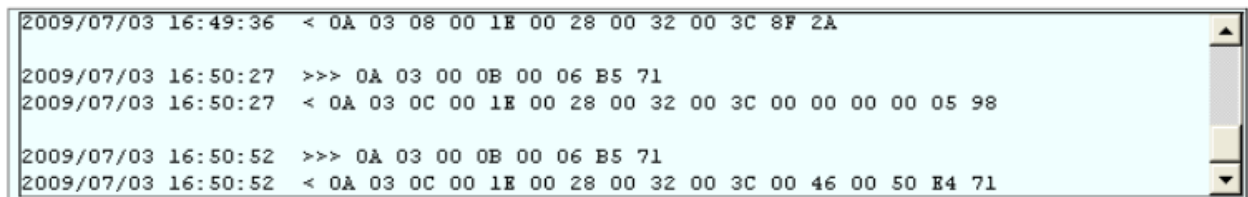
Building the model using information theory requires a finite list of message types and frequencies of those messages. Here, SCADA messages that consist of readings on a continuous scale are converted to messages that either trigger an alarm or do not trigger an alarm. This provides a finite set of possible messages for use in calculating entropy. The conversion from a continuous scale to a finite scale has been used by others to facilitate calculating entropy. For example, Schutzer calculated the entropy of the location of a ship by using the probability of a ship being located in a specific area on a grid map (Schutzer, 1982:17).

To develop and explain the methodology, SCADA messages for a simple 10 component system are simulated using a random number generator in Microsoft Excel. Microsoft Excel 2007 spreadsheets and Visual Basic 6.5 macros are used to format the message data. This data is used to ensure the model works correctly and to develop the application of attack scenarios. JMP 10.0.2 software is used to build the full factorial experiment and analyze the results.

Excel and Visual Basic are used to create the information theory based anomaly detection model. This model calculates the information-theoretic measure of entropy over time using a sliding window. Several attack scenarios are analyzed for their impact to the system, such as changing the alarm distribution of the SCADA system without triggering an anomaly from the model. Chapter IV applies the model and analyzes attack scenarios using data from a water treatment plant.

Data Format

This research begins with the assumption that the user has collected SCADA message traffic from their system and used their own software to format the messages into Excel. It is also assumed that the user has knowledge of the Modbus Map, which identifies the message sending devices on the SCADA system and the operating thresholds. The Modbus Map information is used to create a look up table to determine if a message results in an alarm. (The water treatment plant data used in Chapter IV did not contain the component thresholds; instead they were constructed using known abnormal plant states, which is described that chapter.) The data log is in hexadecimal format, which must be converted to decimal and binary (where applicable) before analysis can begin. Figure 12 (Simply Modbus, 2013) is an example of a data log of Modbus traffic in hexadecimal format.

```
2009/07/03 16:49:36   < 0A 03 08 00 1E 00 28 00 32 00 3C 8F 2A

2009/07/03 16:50:27   >>> 0A 03 00 0B 00 06 B5 71
2009/07/03 16:50:27   < 0A 03 0C 00 1E 00 28 00 32 00 3C 00 00 00 00 05 98

2009/07/03 16:50:52   >>> 0A 03 00 0B 00 06 B5 71
2009/07/03 16:50:52   < 0A 03 0C 00 1E 00 28 00 32 00 3C 00 46 00 50 E4 71
```

Figure 12. Example Modbus Data Log. (Simply Modbus, 2013)

A Google search with the terms "Modbus Excel" returns a multitude of different professional and shareware examples of formatting Modbus data in Excel. An example from a water supply system is shown in Figure 13 (Weidling, 2000). This research is based on the Modbus protocol, but other protocols also have the ability to export messages to a similar Excel format. After the Modbus data is in a spreadsheet, the next step to formatting data can begin.

		City of Arcata Water Supply System				
Date and time of this sheet			Alarms, Levels and Alliance Flow Summary			
1/23/98 16:47						
RUGID SITE #	SITE 3	SITE 4	SITE 6	SITE 6	SITE 6	
TANK # at site	TANK 1AB	TANK 2	TANK 3	TANK 7	TANK 4	
Present Level	**28.8**	**27.8**	**27.8**	**11.0**	**24.2**	
High-High setpoint	31.5	31.0	30.0	18.5	30.5	
Low-Low setpoint	23.0	21.0	22.0	5.0	17.0	
High-High alarm	**OK**	**OK**	**OK**	**OK**	**OK**	
Low-Low alarm	**OK**	**OK**	**OK**	**OK**	**OK**	
Pump One-fail alarm	**OK**	**OK**	**OK**		**OK**	
Pump Two-fail alarm	**OK**	**OK**	**OK**		**OK**	
Power-fail alarm	**OK**	**OK**	**OK**	**OK**	**OK**	
Com-fail alarm	**OK**	**OK**	**OK**	**OK**	**OK**	
Lead Pump Mode	AUTO	AUTO	AUTO		AUTO	
Lag Pump Mode	AUTO	AUTO	AUTO		AUTO	
Site voltatge	13.2	13.9	13.2	13.2	14.6	

Figure 13. Example of SCADA data formatted in Excel. (Weidling, 2000)

Once the message traffic is displayed in Excel, it is formatted for use with the information entropy measure. The messages are formatted to show when the message was sent, the origin of the message, and the alarm state for that device. It is also useful to create a column to label the polling cycle of the message. Table 1 shows an example of the formatted messages using the data from Figure 13. The current reading, which is a continuous value, is converted to a binary alarm state with *1* for alarm and *0* for no alarm. The alarm column compares the current reading to the set threshold values using a look up table for the origin of the message, shown in Table 2. This format is suitable for calculating entropy.

Table 1. Formatted SCADA data.

Time Period	Cycle#	Slave ID	Coil/Register	Reading	Alarm?
1/23/1998 16:47	1	SITE 3	TANK 1AB	28.8	0
1/23/1998 16:47	1	SITE 4	TANK 2	27.8	0
1/23/1998 16:47	1	SITE 6	TANK 3	27.8	0
1/23/1998 16:47	1	SITE 6	TANK 7	11	0
1/23/1998 16:47	1	SITE 6	TANK 4	24.2	0

Table 2. Threshold look up table.

THRESHOLD LOOK UP TABLE		
Origin	Low	High
SITE 3 TANK 1AB	23	32
SITE 4 TANK 2	21	31
SITE 6 TANK 3	22	30
SITE 6 TANK 7	5	19
SITE 6 TANK 4	17	31

Simulated SCADA Message Traffic

To facilitate explanation of the methodology for this research and build a basic anomaly detection model, messages from a simple notional SCADA system are simulated. A SCADA system with 10 slave devices sending messages to the master using a full cycle polling system is simulated using Excel. The 10 devices are given identification numbers 1 through 10. Each of these devices sends a message containing its current sensor reading to the master every cycle, which is once per hour. This is equivalent to Modbus function code 3, reading the register.

Sensor readings were simulated using the Excel random number generator and the normal distribution to represent several notional SCADA components. The normal distribution was used because some of the sensor components from the water treatment plant data, used in Chapter IV, have normally distributed continuous values. Different means and standard deviations were used to represent several SCADA components

49

(origins) that report continuous readings. Here, the values are unitless, but could represent any continuous reading such as pressure or temperature. The values are bounded at zero to prevent negative readings. The threshold values were generated using a range of plus or minus one or two standard deviations from the mean, creating a small number of alarms for each component. The choice for the threshold values are notional and used only for building a working model; it was desired to have alarms for some of the components to ensure the model was working correctly. Table 3 shows the simulated message logs for the first two cycles, formatted for use in the anomaly detection model. Again, *1* signifies an alarm and *0* signifies no alarm. Table 4 shows the look up table used to calculate the alarm column.

Table 3. Simulated SCADA data for creating the anomaly detection model.

Time Period	Cycle#	Origin	Reading	Alarm?
9/1/2013 01:00:00	1	1	54.17	0
9/1/2013 01:00:00	1	2	53.35	0
9/1/2013 01:00:00	1	3	54.96	0
9/1/2013 01:00:00	1	4	85.26	0
9/1/2013 01:00:00	1	5	4.45	0
9/1/2013 01:00:00	1	6	87.71	0
9/1/2013 01:00:00	1	7	874.96	1
9/1/2013 01:00:00	1	8	53.71	0
9/1/2013 01:00:00	1	9	3.36	0
9/1/2013 01:00:00	1	10	47.91	0
9/1/2013 02:00:00	2	1	42.44	1
9/1/2013 02:00:00	2	2	51.71	0
9/1/2013 02:00:00	2	3	66.65	0
9/1/2013 02:00:00	2	4	86.20	0
9/1/2013 02:00:00	2	5	6.13	0
9/1/2013 02:00:00	2	6	105.21	0
9/1/2013 02:00:00	2	7	786.10	0
9/1/2013 02:00:00	2	8	57.57	0
9/1/2013 02:00:00	2	9	2.78	0
9/1/2013 02:00:00	2	10	48.29	0

Table 4. Look up table for alarm thresholds.

	Threshold Look Up Value	
Origin	Low	High
1	43.95	54.95
2	50.02	67.66
3	49.48	68.08
4	70.98	89.04
5	3.21	6.84
6	73.50	105.27
7	736.02	860.72
8	45.28	59.07
9	2.53	3.48
10	45.42	56.27

Anomaly Detection Model

An anomaly detection model was built using the information entropy of SCADA system messages. Entropy measures the uncertainty in knowing which messages will occur next. Changes in entropy can be used to detect anomalies in communication, as discussed by Lee and Xiang (2001). Similar to Winter, Lampesberger, Zeilinger, and Hermann (2011), this model uses entropy over time with a sliding window approach to measure the difference between the predicted entropy and observed entropy. The entropy is predicted using a smoothing algorithm, discussed in the next section. The difference (prediction error), δ (Winter *et al*., 2011:198), between the observed value y_i and predicted value $\hat{y}_i$ is

$$\delta_i(y_i, \hat{y}_i) = |\hat{y}_i - y_i| \tag{15}$$

The prediction error is the same as the absolute value of the residual. If this difference is greater than a value set by the learning data, then an anomaly alarm is triggered. The learning data is a user set number of initial messages to baseline the prediction error, which is used to set the anomaly flag value. This data should be representative of normal

51

system operations and should not include any required warm-up period. The following

subsections describe the details of this model.

Entropy Time Series

The foundation of the anomaly detection model is calculating the system entropy

over time. Once the data is in the format of Table 3, an Excel spreadsheet is used to

calculate the system entropy based on the frequency of alarms for each component,

shown in Figure 14. The user constructs the left side of the figure (shown in green); it

shows all the possible message types of alarm or no alarm for each component. The

Counts column is a sum of all the messages from a particular component (*Origin*) with a

specific alarm status (*Alarm*) that occur within a window of cycles (*Start Cycle* to *Stop*

Cycle). The empirical probability of a message type (*Prob*) is determined using the

number of messages divided by the total number of messages. The *System Entropy* value

is the entropy of the system in that cycle window.

System Entropy	0.858		Start Cycle	1
			Stop Cycle	5
			Messages	50
Origin	**Alarm**	**Counts**	**Prob**	**Bits**
1	0	2	0.040	4.644
2	0	4	0.080	3.644
3	0	5	0.100	3.322
4	0	5	0.100	3.322
5	0	5	0.100	3.322
6	0	4	0.080	3.644
7	0	4	0.080	3.644
8	0	5	0.100	3.322
9	0	5	0.100	3.322
10	0	4	0.080	3.644
1	1	3	0.060	4.059
2	1	1	0.020	5.644
3	1	0	0.000	0.000
4	1	0	0.000	0.000
5	1	0	0.000	0.000
6	1	1	0.020	5.644
7	1	1	0.020	5.644
8	1	0	0.000	0.000
9	1	0	0.000	0.000
10	1	1	0.020	5.644

Figure 14. Entropy calculations.

The information entropy of the system is calculated over a sliding window of time. The entropy is calculated using the probabilities of each message type i and the number of components N within the window. Each component has two message types (alarm or no alarm), therefore the total number of message types is $2N$. Here, $p_i \log p_i = 0$ when $p_i = 0$. (Shannon, 1948: 398). The entropy is normalized by dividing by $\log(2N)$ to simplify interpretation (Winter *et al.*, 2011:196):

$$H = -\frac{\sum_{i=1}^{2N} p_i \log p_i}{\log(2N)} \tag{16}$$

As discussed in Chapter II, the sliding window approach involves calculating entropy over a period and then shifting the widow by one cycle and recalculating the entropy. The widow also overlaps the previous window. For example, a widow size of five cycles means that the entropy is calculated for messages starting in cycle one and

53

ending in cycle five. The next entropy is calculated from cycle two through cycle six. A

Visual Basic macro is used to increment the start cycle and save the entropy for each

window into a new spreadsheet. The user can change the size of the window.

Smoothing Algorithms for Prediction

The model identifies an anomaly when the observed entropy differs from the

predicted entropy by more than a set amount, δ. This research uses two different

prediction algorithms: moving average (MA) and single (a.k.a. simple) exponential

smoothing (SES). At time t ($t = 1, 2, \dots T$) the MA predicted value $\hat{y}_t$ (Krishnamurthy,

Sen, Zhang, & Chen, 2003:237),

$$\hat{y}_t = \sum_{i=1}^{W} \frac{y_{t-i}}{W} \quad \text{for } 1 \leq W \leq T - 1 \tag{17}$$

is the average of the last W observed values y_t. The value of W determines how many of

the previous observations are used to predict the next value. W is an integer that can

range from one (where the prediction is equal to the last observed value) to one less than

the number of entropy values calculated in the baseline. The next section discusses how

W is determined.

The SES algorithm, an exponentially weighted moving average (EWMA), was

first developed by S. W. Roberts (Roberts, 1959) and modified by J. Stuart Hunter;

Hunter's variation of SES is

$$\hat{y}_t = \begin{cases} y_1 & \text{if } t < 2 \\ \alpha y_{t-1} + (1 - \alpha)\hat{y}_{t-1} & otherwise \end{cases} \tag{18}$$

where $\hat{y}_t$ is the predicted value at time period t, for $t = (1, 2 \dots T)$, y_t is defined as the

observed value at time period t, and α is a smoothing parameter that "determines the

depth of memory of the EWMA" and ranges from zero to one (Hunter, 1986:206). The first prediction is initialized using the first observed value. (Hunter, 1986:206) For the notation, Hunter used λ as the smoothing parameter; α is used here to align with recent research using SES, such as Winter *et al.* (Winter, Lampesberger, Zeilinger, and Hermann, 2001:197).

The MA and SES algorithms do not account for seasonal or trend factors in the data (Winter *et al.*, 2001:197). The need for incorporating these factors into the prediction algorithm depends on the source of the data and the window size. MA differs from SES by equally weighting a set number of previous observations into the next predicted outcome. SES weights the previous observations an exponentially decreasing amount by the age of the observation.

Determining Model Parameters

Several model parameters must be set to perform anomaly detection: window size, anomaly flag threshold, baseline size, and the smoothing parameters. These parameters impact the sensitivity to detecting attacks and the number of false alarms reported. The objective is to maximize the number of manipulations detected and minimize the false alarms.

The size of the sliding window is an important aspect of anomaly detection. Smaller window sizes are more sensitive to changes, but can cause more false positives; larger windows result in less false positives, but may miss more anomalies (Winter *et al.*, 2011:199). This research investigated the impact of varying the window size on attack detection percentage and false positive percentage (also known as false positive rate). The user sets the sliding window size on the model interface.

The anomaly flag threshold is set using the amount of difference, δ, between the observed entropy and predicted entropy. Winter *et al.* allowed the user to set this threshold (Winter *et al.*, 2011:199). In this model, the anomaly threshold is set automatically using the baseline data. The threshold is assigned the maximum δ value from the baseline. The choice of this method is from assuming the baseline data is representative of normal operations with no attacks or system problems present. This method causes the baseline data to have no anomaly flags.

The baseline size determines the number of cycles that make up normal operations. The baseline data is important because it sets the threshold for anomaly flags in the detection model. Since this research has a limited amount of data, the baseline is chosen to balance establishing a proper threshold and still have enough remaining message data to investigate different alarm scenarios. Additionally, the baseline should be representative of normal operating conditions and contain no attacks or system problems. The user can set the baseline size as an input to the model. If the data has a start up period that is different from normal operations then the baseline should begin after the start up period is over. The simulated data used in this chapter and the data used in Chapter IV do not have a start up period; therefore, the baseline begins with the first cycle.

The smoothing parameters for MA and SES forecasting algorithms are assigned by choosing the value for W and α, respectively, that minimizes the square root of the Mean Squared Forecast Error (MSFE). The MSFE,

$$MSFE = \frac{1}{T}\sum_{i=1}^{T}(\hat{y}_i - y_i)^2 \qquad (19)$$

is the average of the squared residuals, which also equals the average of the prediction

errors (Gelper, Fried, & Croux, 2010:7). This method selects smoothing parameters that

minimize the amount of error in the predictions overall. It also equally weights the

residuals and is sensitive to outliers. This could be an issue if "One very large forecast

error causes an explosion of the MSFE, which typically leads to smoothing parameters

being biased towards zero" (Gelper *et al.*, 2010:7). The user can optimize the smoothing

parameters for the model by selecting a button on the model interface that runs a Visual

Basic macro. The macro runs the solver analysis tool to minimize MSFE by changing the

smoothing parameter. The default Excel solver settings were used, which applies the

Generalized Reduced Gradient Algorithm. The results of the solver can be checked by

plotting the MSFE against the possible values for each smoothing parameter. The

following are the three constraints for the moving average constant, W: it must be an

integer, it must be greater than 0, and it is limited to 10% of the number of baseline

cycles. The third constraint is in place so that the MSFE is not biased to a small value in

the event that the errors are small for the last few baseline cycles. The constraints for α

are that it must be between zero and one. In the event that W and α both equal one, the

prediction algorithms become the same and the predicted value is equal to the last

observed value.

Anomaly Detection

The model flags a cycle for containing an anomaly when the observed entropy differs from the forecasted entropy by an amount greater than the anomaly flag threshold. The anomaly flag points to the cycle recently added to the sliding window. The flagged anomaly could mean an attack has occurred during that cycle, a problem with the system, or by normal system fluctuations that caused a false positive. A potential issue with this definition is that attack manipulations may not immediately trigger an anomaly from the model, but they could impact the entropy at a later point and set off an anomaly flag. This possible lag between the attack and the anomaly flag may cause the operator to search for attacks in the wrong cycle; this is an area for future research. As discussed, for this research, the anomaly flag threshold is set using the baseline messages. The baseline is also used to optimize the smoothing parameters. The same smoothing parameters are used for forecasting the entropy values of the data under analysis.

Figure 15 shows an example of an anomaly flag generated when the difference in forecast value and observed value is greater than the threshold (shown here as *Threshold*). The anomaly occurred when the sliding window started at cycle 15. With a window size of two cycles, this means that cycle 16 may have a system problem or attack. The model points to cycle 16, shown in the *Flagged Cycle* column in the figure, because it was the newest cycle added to the sliding window and the data in that cycle caused an entropy change larger than the threshold. This example shows that both the MA and SES algorithms flagged the anomaly.

System Entropy Over Time				Moving Average		Simple Expon. Smooth		
Window Size: 2				**Baseline Max** 0.004		**Baseline Max** 0.004		
Start	**Entropy**	**MA**	**SES**	**MA Diff**	**Anomaly?**	**SES Diff**	**Anomaly?**	**Flagged Cycle**
13	0.840	#N/A	#N/A					
14	0.840	0.840	0.840	0.000		0.000		
15	0.861	0.840	0.840	0.021	Anomaly	0.021	Anomaly	16
16	0.861	0.861	0.857	0.000		0.004		

Figure 15. Anomaly detection output and example of an anomaly flag.

The anomaly detection model classifies each cycle into one of four categories: true positive, false positive, false negative, and true negative. An attack or problem is considered detected, and falls into the true positive (TP) category, if an anomaly flag points to a cycle containing an attack manipulation or problem. The model crosschecks the flagged cycles against the known problems and applied attacks. An anomaly is considered a false positive (FP) if there are no attack manipulations or known problems occurring during that cycle. If the model does not flag a known attack or problem, then it is considered a missed anomaly, also known as a false negative (FN). The last category, true negative (TN), occurs when a cycle that does not contain an attack or known problem is not flagged for an anomaly. The next sections discuss attack analysis in more detail.

Attack Scenario Experiment

The second objective of this research is to analyze the types of attack scenarios detected by the model and their impact on false positives. Here the term *attack* is a scenario of alarm manipulations. Alarm *manipulations* include adding or removing alarm messages from components. An experiment was designed using a full factorial combination of certain attack scenario possibilities. The experiment was repeated using

two different sliding window sizes for the anomaly detection model. The response

variables are the true positive percentage (TP%) and the false positive percentage (FP%)

for the two forecasting algorithms. TP% is also known as sensitivity or the true positive

rate; FP% is also known as the false positive rate (Linda, Vollmer, and Manic,

2009:1832). The goal is to maximize the TP% and minimize the FP%. The next section

provides the details on the experimental analysis. This analysis provides insight into the

impact of different attack scenarios and changing the sliding window size.

Attacks are added to the model and the entropy over time is calculated using the

same calculations described previously. An attack matrix, shown in Figure 16, creates

manipulations of the SCADA alarms. The dark shaded region on cycles 1-12 shows the

baseline data. The baseline data does not contain manipulations. Alarms are added or

removed by entering a *1* in the corresponding cycle and origin (component) of the attack

matrix, as shown in the figure. The alarm column shown in Table 3 changes based on the

attack matrix. If the original data did not have an alarm for a cycle and origin, then a

lookup function uses the *Added Alarms* attack matrix (left side of figure) to determine if a

manipulation has occurred and if an alarm is added. If the original data did have an

alarm, then the lookup function uses the *Removed Alarms* attack matrix (right side of

figure) to determine if the alarm has been removed. Once the attack scenario is applied

to the attack matrices, the user runs a macro to calculate the entropy over time with the

manipulated alarm messages. The output for anomaly detection is shown in Figure 15.
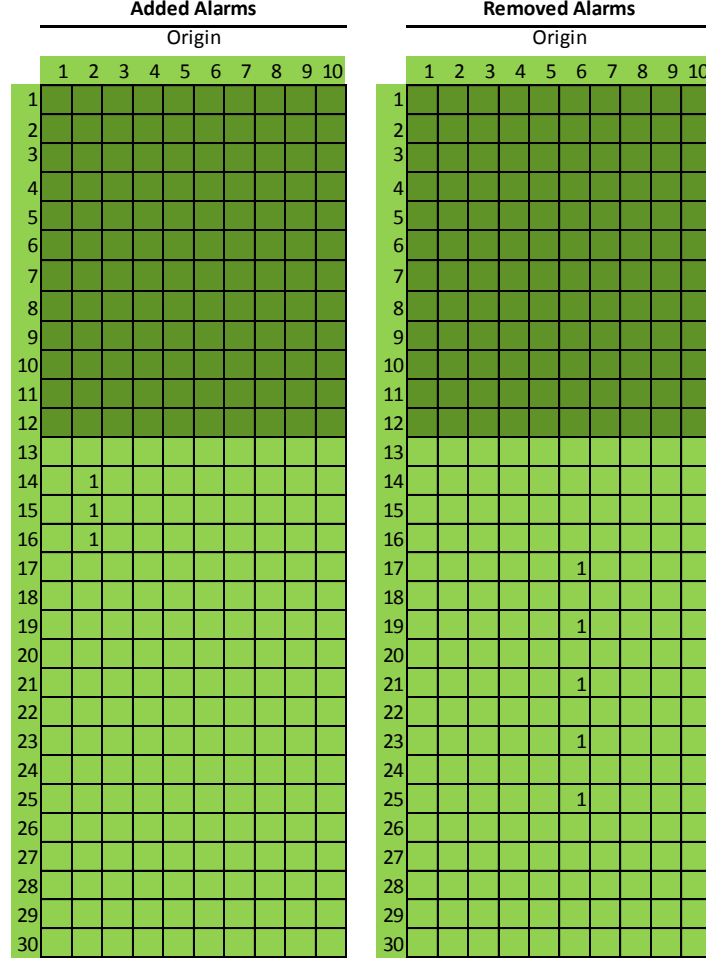
Figure 16. Attack matrices used for alarm manipulation.

The response variables are the true positive percentage (TP%) and false positive

percentage (FP%) for each forecasting algorithm, which are derived from the confusion

matrix shown in Figure 17 and the following equations (Linda, Vollmer, and Manic,

2009:1832):

$$TP\% = \frac{TP}{TP + FN}$$

$$FP\% = \frac{FP}{FP + TN}$$

(20)

The TP% is the number of detected cycle attacks and known cycle problems (true positives) divided by the total number of cycle attacks and known problems (true positives and false negatives). In other words, the TP% is the percent of attacks/problems detected out of the total number of attacks/problems. The FP% is the number of false positives out of the number of normal states; it is the number of anomalies reported that do not correspond to any manipulations or known problems (false positives) divided by the number of cycles that do not contain problems or attacks. The response variables are computed on a separate spreadsheet titled *Summary*. The *Summary* sheet lists all the attack manipulations from the attack matrices and the anomalies reported by each forecasting method. This sheet also displays the confusion matrix.

		Predicted	
		"Attack/Problem"	"No Attack/Problem"
Actual	Attack/Problem	True Positive (TP)	False Negative (FN)
	No Attack/Problem	False Positive (FP)	True Negative (TN)

Figure 17. Confusion matrix.

The experimental design is composed of five attack scenario factors and one factor concerning the model. Five factors have two levels and one factor has three levels. Table 5 shows the factors, the type of data (ordinal or categorical), and description of the levels. The *Number of Cycles* (NC) is the number of cycles that have alarms added or removed in an attack; its levels include a low and high number of cycles with manipulations. The *Number of Targets* (NT) factor is the number of components attacked; its levels are low and high number of components. The *Attack Spacing* (AS) is the number of cycles between the groups of manipulations, with levels of low and high number of cycles. The *Attack Distribution* (AD) includes grouping the manipulations

62

together, distributing them over a period of time, or increasing them from a small number to a larger number over time. Increasing manipulations may have the impact of changing the entropy over time in small increments, therefore going undetected. The *Attack Type* (AT) has two different levels: adding alarms and removing alarms. The last factor changes the detection model and is not part of the attack scenario. The *Window Size* (WS) has the levels of small and large sliding window size.

Table 5. Experiment factors and levels.

Factor	Type	Levels		
Number of Cycles (NC)	Ordinal	Low	High	
Number of Targets (NT)	Ordinal	Low	High	
Attack Spacing (AS)	Ordinal	Low	High	
Attack Distribution (AD)	Categorical	Grouped	Distributed	Increasing
Attack Type (AT)	Categorical	Add	Remove	
Window Size (WS)	Ordinal	Small	Large	

This methodology uses a full factorial design with five replicates. A full factorial design includes all possible combinations of each factor at all levels in every experimental run; this design is considered "most efficient" for analyzing the effects of several factors (Montgomery, 2011:183). Replicates are used to "obtain an estimate of the experimental error" and increase the possibility of getting statistical significance of small differences (Montgomery, 2011:12). Each replicate has $2 * 2 * 2 * 3 * 2 * 2 = 96$ samples. Replicates are generated by increasing the cycle where the attack manipulations begin. The original data does not change for each experimental run, but the distribution of alarms varies from cycle to cycle. The attack start cycle is increased the same way when the experiment is repeated for different window sizes. The possible number of replicates depends on the amount of data available, the window size, and the amount the

63

attack start cycle is increased for each replicate. Eventually, shifting the attack start cycle

will not allow the full attack to be applied to the remaining data. The model does not

randomize the samples since the original data is the same for each trial.

Experiment Analysis

       The results are analyzed using an analysis of variance (ANOVA) and the Kruskal-

Wallis (K-W) test, when needed, with JMP software. The first test is conducted to

determine if there is a relation between the response variables and the design factors; also

known as the treatments (Kutner, Nachtsheim, and Neter, 2004:226). This test uses the

means model:

$$y_{ij} = \mu_i + \epsilon_{ij} \quad i = 1,2, \dots, a; j = 1,2, \dots, n \tag{21}$$

where y_{ij} is observation j of factor level i (for n observations and a different factor

levels), μ_i is the mean of level i, and ϵ_{ij} is the random error component (Montgomery,

2011:69).

       ANOVA is used to test the hypothesis that the mean response for each level of a

factor (treatment) are equal (Montgomery, 2011:74):

$$H_o: \mu_1 = \mu_2 = \cdots = \mu_a$$
$$\tag{22}$$
$$H_1: \mu_i \neq \mu_j \quad \text{for at least one } ij \text{ pair}$$

The alternative is that at least one mean response is different. The test statistic, F_o,

$$F_o = \frac{SS_{Treatments}/(a-1)}{SS_{Error}/(N-a)}$$

$$SS_{Treatments} = \frac{a}{n}\sum_{i=1}^{a} y_{i.}^2 - \frac{y_{..}^2}{N}$$

(23)

$$SS_{Total} = \sum_{i=1}^{a}\sum_{j=1}^{n_i} y_{ij}^2 - \frac{y_{..}^2}{N}$$

$$SS_{Error} = SS_{Total} - SS_{Treatments}$$

is the mean of the sum of the squared values for each treatment ($SS_{Treatments}$) divided by

the mean of the sum of squares for the error (SS_{Error}) (Montgomery, 2011:74). Here,

there are a different treatments, N total observations, n_i observations for treatment i, $y_{..}$ is

the total of all observations, and $y_{i.}$ is the total for treatment i. The test statistic "is

distributed as F with $a-1$ and $N-a$ degrees of freedom" (Montgomery, 2011:74).

The main effects and interaction of factors are considered statistically significant if the p-

value from the F statistic is less than an alpha of 0.05. The p-value is found with JMP

software.

In addition to the p-value from the F statistic, the percent contribution to the total

sums of squares is computed for each factor and full factorial interaction. "The

percentage contribution is often a rough but effective guide to the relative importance of

each model term" (Montgomery, 2011:246). It is the sum of squares for a factor divided

by the total sum of squares.

There are assumptions necessary to use the ANOVA that need to be checked.

First, in order to avoid false positive results using ANOVA, the residuals should be

normally distributed. "In general, moderate departures from normality are of little

concern in the fixed effects analysis of variance [and it]…is robust to the normality assumption" (Montgomery). This research will use a normal quantile plot of the residuals to assess the normality assumption.

Another assumption is that the data is independent and does not contain any significant outliers (Montgomery, 2011:82). The samples are collected independently as described earlier in this chapter when the replicates were defined. This research assumes that there are no outliers in the data used in Chapter IV. Large residuals are assumed to be legitimate values caused by the variation in the SCADA data.

The final assumption used in this research is that the replicates have constant variance. This is assessed by plotting the residuals against the predicted value. This plot should show no structure. The residuals should not reveal any pattern and should have a constant variance.

If the residuals do not appear normally distributed, the K-W test will be used to confirm the ANOVA results. This is a non-parametric ranks test which Douglas Montgomery states can be used in the event of nonnormality:

> When we are concerned about the normality assumption or the effect of outliers or "wild" values, we recommend that the usual analysis of variance be performed on both the original data and the ranks. When both procedures give similar results, the analysis of variance assumptions are probably satisfied reasonably well, and the standard analysis is satisfactory. When the two procedures differ, the rank transformation should be preferred because it is less likely to be distorted by nonnormality and unusual observations. (Montgomery, 2011:130)

The K-W test begins by rank-ordering the results for each treatment; ties are given the average value of the ranks of the ties group. This test "is used to test the null hypothesis that the *a* treatments are identical against the alternative hypothesis that some

of the treatments generate observations that are larger than the others" (Montgomery, 2011:128). The hypothesis shown mathematically is

$$H_o: \frac{R_1}{n_1} = \frac{R_2}{n_2} = \cdots = \frac{R_a}{n_a}$$

$$H_1: \frac{R_i}{n_i} \neq \frac{R_j}{n_j} \quad \text{for at least one } ij \text{ pair}$$

(24)

The K-W test statistic is

$$H = \frac{1}{S^2} \left[\sum_{i=1}^{a} \frac{R_i^2}{n_i} - \frac{N(N+1)^2}{4} \right]$$

(25)

(26)

$$S^2 = \frac{1}{N-1} \left[\sum_{i=1}^{a} \sum_{j=1}^{n_i} R_{ij}^2 - \frac{N(N+1)^2}{4} \right]$$

where R_i is the "sum of the ranks of the ith treatment...n_i is the number of observations of the ith treatment, N is the total number of observations", and S^2 is the variance (Montgomery, 2011:129). For large value of n_i, H is approximately distributed as a chi squared with $\alpha - 1$ degrees of freedom. The null hypothesis is rejected if

$$H > \chi^2_{\alpha,\alpha-1}$$

(27)

When the null hypothesis is rejected, it means that there is enough evidence to conclude that at least one of the attack factors or interactions had a significant impact on the detection rate. JMP software is used to compute the p-value for the test statistic. (Montgomery, 2011:129)

Summary

This chapter explained the methodology for the anomaly detection model and the experiment for analyzing the attack scenarios. The model was developed using a notional SCADA system and applying information entropy and a prediction algorithm. Chapter IV applies the anomaly detection model to a publicly available dataset of water treatment plant SCADA messages. This dataset is used to evaluate the model and analyze the significance of different attack scenarios applied to the dataset.

IV. Analysis and Results

Introduction

This research applies the information entropy anomaly detection model and attack scenarios to data from a water treatment plant monitoring system. A 2007 report to Congress from the US Government Accountability Office (GAO) identified water treatment as a critical infrastructure and its security as a national priority. "Critical infrastructures are physical or virtual systems and assets so vital to the nation that their incapacitation or destruction would have a debilitating impact on national and economic security, public health, and safety" (GAO, 2007:3). GAO described the vulnerability and threats to the Supervisory Control and Data Acquisition (SCADA) system:

> Critical infrastructure control systems face increasing risks due to cyber threats, system vulnerabilities, and the serious potential impact of attacks as demonstrated by reported incidents. Threats can be intentional or unintentional, targeted or nontargeted, and can come from a variety of sources including foreign governments, criminal groups, and disgruntled organization insiders. Control systems are more vulnerable to cyber attacks than in the past for several reasons, including their increased connectivity to other systems and the Internet. (GAO, 2007:2)

The GAO report discussed several examples of attacks on water treatment systems. In 2000, a disgruntled applicant for a government job in Australia reportedly "[broke] into the controls of a sewage treatment system…altered data for…pumping stations and caused malfunctions in their operations, ultimately releasing about 264,000 gallons of raw sewage into nearby rivers and parks" (GAO, 2007:15). Another example was in 2006 in Harrisburg, Pennsylvania when "a foreign hacker penetrated security at a water filtering plant…[and] planted malicious software that was capable of affecting the plant's water treatment operations" (GAO, 2007:16-17).

The scenario for this chapter is the analysis of defending a water treatment plan from an attacker. Here, the water treatment plant is employing the entropy based anomaly detection model defined in Chapter III to detect problems with the system and identify alarm manipulation attacks. The objectives of this chapter are to evaluate the use of the detection model on this water treatment system and identify the impact of different attack scenarios on detection rate and false positive percentage. Analysis of variance (ANOVA) is used to evaluate the factors specific to the model and the attack scenario factors. This process is repeated on a modified dataset that contains no abnormal cycles with the exception that only added attacks are considered in the analysis of the modified dataset. Appendix A contains step-by-step instructions for running the model in Excel.

Data

Complete message traffic data and system information from a SCADA system were not available for this research. System owners may be reluctant to provide this information due to the risk of exposing critical vulnerabilities and affecting their system (Chunlei, Lan, & Yiqi, 2010:342). As an alternative, this research uses a publicly available database comprised of daily sensor readings and classification of plant state at a wastewater treatment plant to meet the objectives of evaluating the anomaly detection model using different attack scenarios.

The database used in this project is from the UCI Machine Learning Repository (Bache & Lichman, 2013). The data has 527 days of readings (the samples) from 38 sensors (the system components) covering a period between January 1990 and October 1991, though the database does not cover every day. The original source is the

Autonomous University of Barcelona, Spain, June 1993. Refer to Appendix B for a thorough description of this database, including descriptions of the sensors and plant states.

The database had many missing sensor values, which were addressed before beginning analysis. The sensor data was missing 591 values, consisting of about 3% of the total data. Some sensor readings were missing over 10% of their values and some of the samples were missing more than one sensor reading. The last observation carried forward method of imputation was used because of the source of the data (Young, Weckman, & Holland, 2010:19). Some data acquisition systems respond to missing values by using the "last reported parameter value" (Bentley Systems Inc, 2004:10).

The dataset did not contain information on the alarm thresholds for each sensor. Normally, a SCADA operator would know the alarm threshold settings for each component and be able to adjust those settings (Shaw, 2006:158-159). In order to use the dataset in this research, alarm thresholds for each component (sensor) needed to be derived. This was done using the anomaly detection model as described in the next section.

Setting Model Parameters

Several model parameters were set before the analysis could begin. The alarm threshold values, baseline cycles, smoothing constants, anomaly threshold, and sliding window size all needed to be determined.

Since the dataset contained information about the plant state for each day, the days with abnormal plant states were used to set the alarm threshold. It was assumed that

any reading greater than a certain number of standard deviations from the mean was outside of normal operating thresholds and therefore triggered an alarm. The standard deviation was chosen because many of the distributions fit the normal distribution using the Kolmorgov-Smirnov test with an alpha equal to 0.01 significance, and all of the distributions were mound shaped except for 14 sensors that were bounded on one side. For the 14 components that were bounded, alarms only occurred for values below/above the set number of standard deviations from the mean, depending on which side the data was bounded. Thresholds between three and six standard deviations were evaluated.

The procedure for setting the component thresholds began with identifying the number of cycles for the baseline. The first abnormal state of the water treatment plant occurred at cycle 60; therefore the first 45 cycles were used for the baseline. 45 cycles were used to allow for a buffer of normal cycles to occur in the data under analysis before the first abnormal state occurred. The next step in setting the component thresholds is to calculate the true positive percentage (TP%) and false positive percentage (FP%) for different sizes of the sliding window. For reference, the confusion matrix and calculations for TP% and FP% are shown in Figure 18 and equation (28), respectively (Linda, Vollmer, and Manic, 2009:1832).

| | | Predicted | |
		"Attack/Problem"	"No Attack/Problem"
Actual	Attack/Problem	True Positive (TP)	False Negative (FN)
	No Attack/Problem	False Positive (FP)	True Negative (TN)

Figure 18. Confusion matrix.

72

$$TP\% = \frac{TP}{TP + FN}$$

$$FP\% = \frac{FP}{FP + TN}$$

(28)

This is repeated for alarm threshold values of three, four, five, and six standard deviations from the mean.

The next step in setting the component thresholds is determining the smoothing parameters used in the model. This entails calculating the entropy over time of the baseline cycles using different sliding window settings. The model interface provides the user the option to optimize the smoothing constants using a macro that runs the Excel solver tool, described in Chapter III. Here, every window size and standard deviation setting evaluated resulted in a moving average interval and simple exponential smoothing (SES) constant alpha equal to 1.0. When the smoothing constants are both equal to 1.0, the forecasting algorithms both predict using only the last observed entropy; this is the setting used for the factorial experiment. An additional smoothing constant setting of alpha equal 0.8 is used in setting up the model to compare the performance.

The next step is performed to determine the standard deviation values for the data and the window sizes to be used in the factorial experiment. To do this, the TP% and FP% were calculated for window sizes between two and 15 and for standard deviation values between three and six. This step was repeated for alpha values of 1.0 and 0.8 and using two ways to calculate the prediction error: absolute value of observed minus predicted and only the positive values of observed minus predicted. The goal was to achieve a 100% detection (TP%) of the abnormal cycles at the lowest FP%. This was chosen because most SCADA operators prioritize detecting 100% of system problems

and accept a trade-off of more false alarms, though this depends on the criticality of the system and the cost of responding to false alarms (Gerard, 2005:3). A water treatment plant is considered a critical resource that undetected problems could cause severe consequences (GAO, 2007:3).

The results of this evaluation are shown in Table 6. The anomaly detection model was able to detect 100% of the system problems. All combinations that achieved a 100% TP% are highlighted in green. Of those, the settings achieving the lowest FP% are highlighted in orange. The results showed that an alarm standard deviation (*StDev*) setting of five, a window size (*WS*) of 11, an alpha of 1.0, and using only positive values of prediction errors (*Max Diff Only*) resulted in the lowest FP% for the combinations that achieved a 100% TP%. Here, the model was optimized with this data using the maximum difference setting. Using this setting, only increases in entropy will be detected. Since it is likely that the baseline of the system will contain relatively few alarms sent to the operator, only attacks that add alarm messages and increase the overall entropy will be detected. Removing alarms will not be detected because they would cause the entropy to decrease. Further testing is necessary to determine if this result can be generalized to other systems. Additionally, future research should evaluate using the absolute value of the prediction error to determine if removing alarm attacks can be detected.

Table 6. TP% and FP% for different model parameters.

WS	StDev	Alpha = 1 Abs Value of Diff		Alpha = 1 Max Diff Only		Alpha = 0.8 Abs Value of Diff		Alpha = 0.8 Max Diff Only	
		TP%	FP%	TP%	FP%	TP%	FP%	TP%	FP%
2	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.47%	0.00%	0.00%
3	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.47%	0.00%	0.00%
4	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.47%	0.00%	0.00%
5	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.94%	0.00%	0.00%
6	3	0.00%	0.23%	7.14%	0.23%	0.00%	0.94%	14.29%	0.00%
7	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.70%	0.00%	0.00%
8	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.70%	0.00%	0.00%
9	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.70%	0.00%	0.00%
10	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.94%	0.00%	0.00%
11	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.94%	0.00%	0.00%
12	3	0.00%	0.23%	0.00%	0.00%	0.00%	0.70%	0.00%	0.00%
13	3	0.00%	0.23%	7.14%	0.00%	14.29%	0.70%	21.43%	0.00%
14	3	14.29%	1.17%	14.29%	0.00%	21.43%	2.34%	21.43%	0.00%
15	3	0.00%	0.23%	0.00%	0.00%	7.69%	0.93%	7.69%	0.00%
2	4	64.29%	4.22%	50.00%	1.41%	71.43%	5.39%	57.14%	1.64%
3	4	57.14%	5.62%	50.00%	2.58%	57.14%	6.09%	50.00%	2.11%
4	4	42.86%	5.85%	42.86%	2.58%	50.00%	6.56%	50.00%	2.81%
5	4	42.86%	6.32%	42.86%	2.11%	57.14%	7.49%	57.14%	2.34%
6	4	64.29%	6.56%	64.29%	2.58%	71.43%	7.26%	71.43%	2.58%
7	4	57.14%	4.92%	57.14%	1.64%	64.29%	6.09%	64.29%	1.87%
8	4	50.00%	5.62%	50.00%	2.11%	64.29%	7.03%	64.29%	2.58%
9	4	57.14%	6.09%	57.14%	2.11%	64.29%	7.26%	64.29%	2.58%
10	4	64.29%	5.85%	64.29%	2.11%	57.14%	6.79%	57.14%	2.11%
11	4	57.14%	5.39%	57.14%	1.87%	57.14%	6.56%	57.14%	2.58%
12	4	64.29%	5.62%	64.29%	1.87%	71.43%	7.49%	78.57%	2.81%
13	4	57.14%	5.15%	57.14%	1.41%	71.43%	7.03%	85.71%	3.75%
14	4	57.14%	9.84%	57.14%	3.98%	78.57%	11.94%	78.57%	4.68%
15	4	50.00%	5.15%	50.00%	1.41%	50.00%	6.32%	50.00%	1.64%
2	5	71.43%	6.32%	57.14%	2.34%	64.29%	10.77%	50.00%	3.51%
3	5	57.14%	9.37%	50.00%	3.98%	71.43%	11.71%	64.29%	4.45%
4	5	71.43%	7.73%	71.43%	3.04%	71.43%	13.35%	71.43%	5.15%
5	5	78.57%	8.90%	78.57%	2.81%	78.57%	12.88%	78.57%	4.68%
6	5	78.57%	7.03%	78.57%	1.87%	92.86%	11.48%	92.86%	4.22%
7	5	78.57%	6.79%	78.57%	1.64%	92.86%	11.01%	92.86%	3.75%
8	5	78.57%	7.96%	78.57%	2.11%	92.86%	13.58%	92.86%	4.68%
9	5	85.71%	8.20%	85.71%	2.81%	92.86%	13.11%	92.86%	4.92%
10	5	92.86%	7.03%	92.86%	2.11%	92.86%	12.41%	92.86%	4.45%
11	5	100.00%	10.54%	100.00%	3.51%	100.00%	13.11%	100.00%	4.45%
12	5	92.86%	7.03%	92.86%	1.41%	100.00%	13.11%	100.00%	5.39%
13	5	92.86%	8.20%	92.86%	2.58%	92.86%	13.11%	92.86%	5.62%
14	5	100.00%	8.20%	100.00%	10.30%	92.86%	10.77%	100.00%	11.71%
15	5	85.71%	7.26%	85.71%	2.58%	85.71%	11.01%	85.71%	3.04%
2	6	92.86%	12.88%	78.57%	5.62%	85.71%	17.80%	71.43%	7.26%
3	6	85.71%	12.65%	64.29%	5.39%	92.86%	16.63%	78.57%	6.56%
4	6	85.71%	11.48%	78.57%	4.22%	78.57%	18.74%	78.57%	7.03%
5	6	85.71%	14.75%	85.71%	6.09%	92.86%	19.20%	92.86%	7.03%
6	6	78.57%	13.35%	78.57%	5.62%	92.86%	18.97%	92.86%	7.03%
7	6	92.86%	12.41%	92.86%	4.68%	100.00%	17.80%	100.00%	6.09%
8	6	85.71%	15.22%	85.71%	6.56%	92.86%	19.91%	92.86%	7.73%
9	6	92.86%	14.99%	92.86%	5.62%	100.00%	20.37%	100.00%	7.96%
10	6	92.86%	17.56%	92.86%	6.79%	92.86%	19.67%	92.86%	7.03%
11	6	100.00%	18.03%	100.00%	6.79%	100.00%	18.97%	100.00%	7.03%
12	6	100.00%	14.05%	100.00%	5.62%	100.00%	18.74%	100.00%	7.03%
13	6	100.00%	17.56%	100.00%	6.79%	100.00%	19.91%	100.00%	8.20%
14	6	100.00%	16.63%	100.00%	6.56%	100.00%	20.14%	100.00%	7.96%
15	6	85.71%	13.82%	85.71%	5.85%	85.71%	17.80%	85.71%	6.79%

The results of the TP% and FP% using the maximum prediction difference method for all standard deviations tested are shown in Figure 19. The arrows in the figure point to the two sliding window sizes used during the factorial experiment. The window size of 11 and standard deviation of five (green line in the graph) had the lowest FP% of all the settings achieving 100% TP%. A window size of four was chosen to compare the impact of the attack scenarios on TP% and FP% for a smaller window size. Now that the data and model parameters are set, analysis of the data can begin.
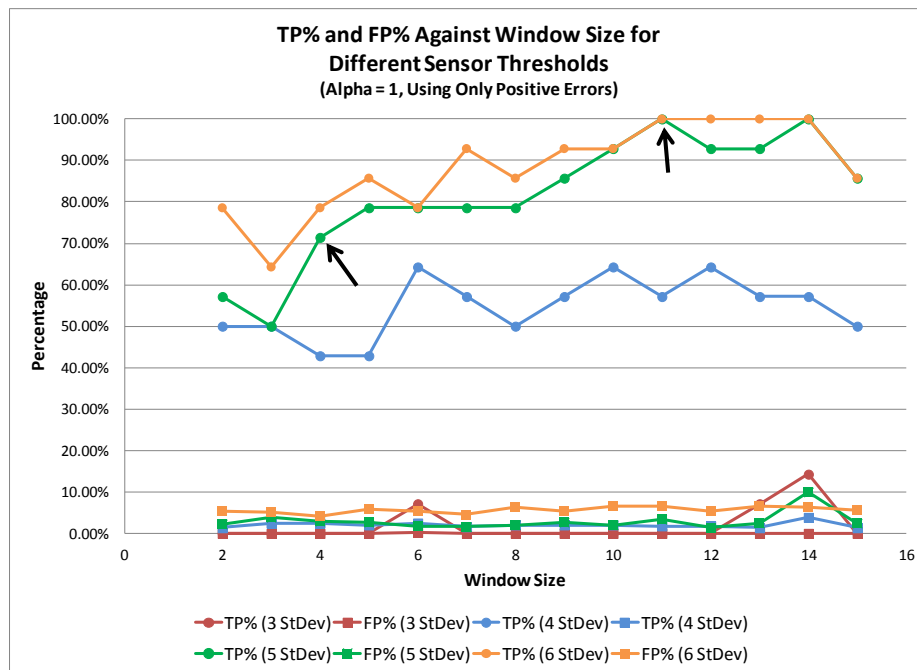
Figure 19. Graph used for setting model parameters.

Model Evaluation

The first objective of the analysis is to evaluate the use of the entropy based anomaly detection model with realistic SCADA data from a water treatment plant. This objective answers the question: Can the model successfully detect system problems

when no attacks are present? The previous section showed the success of the model in

detecting system problems when it was used to set the component alarm thresholds. This

process revealed that 100% of the 14 abnormal cycles were detected when fewer alarms

were present. In fact, there were several different parameter settings that resulted in

100% TP%, as shown in Table 6.

The model performed better when there were less alarms present in the data. The

water treatment data has 326 alarms present at five standard deviations from the mean.

Fewer alarms means that the data has less variation in entropy values and results in a

lower anomaly flag threshold. Figure 20 shows a plot of the prediction error against

cycle when fewer alarms are present. The first 45 cycles are the baseline used to set the

anomaly threshold, shown with an arrow and text in the figure. The red squares represent

the abnormal states and are located at the anomaly threshold value. As shown in the

graph, prediction errors greater than the threshold occur for each abnormal state. These

settings resulted in 100% TP% and 3.40% FP%.

The model performed poorer when there were more alarms present in the data.

There are 1508 alarms in the data at three standard deviations from the mean; nearly five

times as many than the five standard deviation setting. The prediction error has more

variation with more alarms. Figure 21 shows the plot of prediction error against cycle

when there are more alarms. Here, the first 45 cycles, that are the baseline, contain

prediction errors greater than the rest of the data; therefore, the anomaly threshold is too

large and the model is not sensitive to the abnormal cycles. The result of these settings

are a zero TP% and a zero FP%. This result may be specific to this dataset; it is possible

that the baseline data for this dataset happens to have more variance at a lower alarm

threshold than the cycles under evaluation.
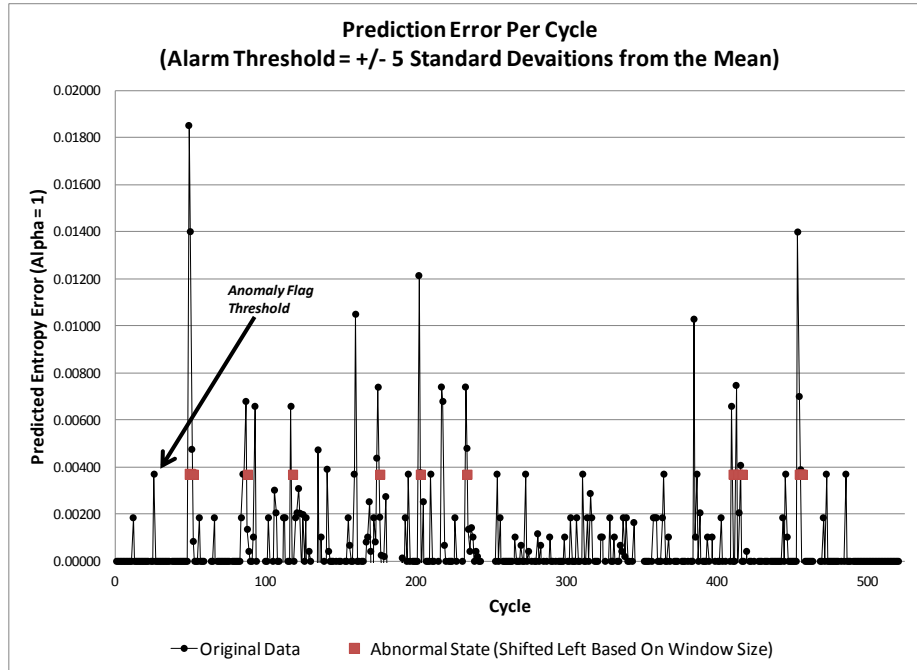
Figure 20. Prediction error against cycle for data with fewer alarms.
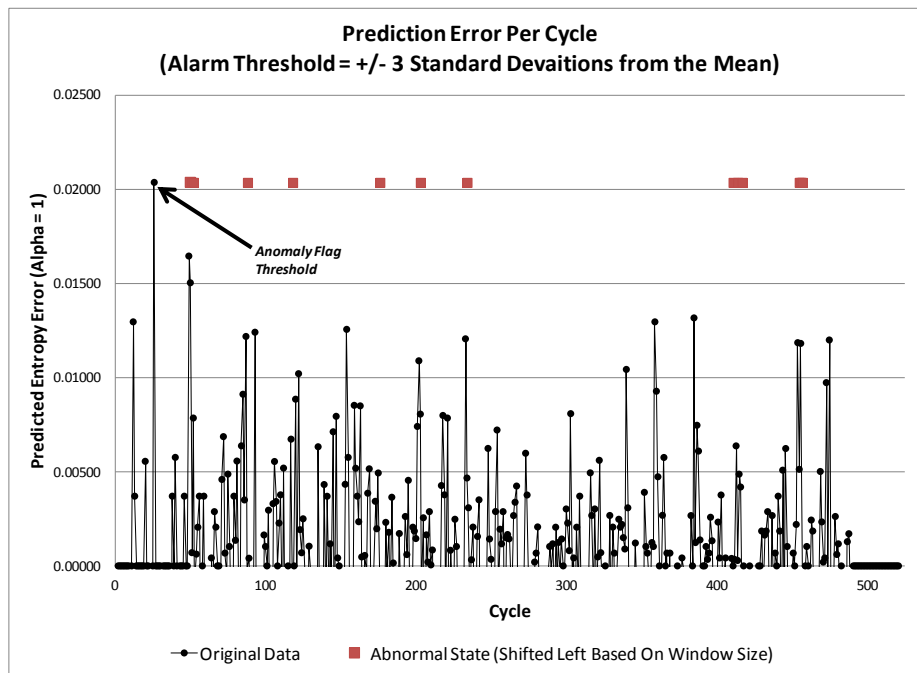
Figure 21. Prediction error against cycle for data with more alarms.

These results may indicate that an entropy based anomaly detection model performs better on a more stable system when fewer alarms are present. After adjusting the model settings, the model was able to detect 100% of the known system problems when no attacks were present. These settings are representative of what a system owner would likely use to detect problems. This research also reveals that the performance of the model is sensitive to the alarm thresholds. Noisier systems produce higher anomaly thresholds and the system problems do not stand out from the noise. The next section evaluates the ability of the model to detect attacks.

Factorial Experiment

The second objective of the analysis is to determine which attack scenario factors and model parameters have a significant impact on TP% and FP%. This research uses a full factorial experiment to evaluate the factors using the optimal window size of 11 cycles. The experiment is repeated for the smaller window size of four cycles and the results are compared. The performance of the model is evaluated using an alpha of 1.0, which is equivalent to using the last observed value as the prediction. The experiment was first run on the original water treatment data with known abnormal states. Next, the experiment was run on a modified dataset that has the abnormal states removed; this allows a direct evaluation of the impact of the attack scenario factors. The experiment is a full factorial design with five replicates. Table 7 shows the values used for each factor level. Details on the design and a description of each level are in Chapter III. Each replicate generates 96 samples. The response variables are the TP% and FP%.

Table 7. Values for the continuous factor levels.

Factor	Type	Levels		
Number of Cycles (NC)	Ordinal	1	6	
Number of Targets (NT)	Ordinal	1	5	
Attack Spacing (AS)	Ordinal	1	5	
Attack Distribution (AD)	Categorical	Grouped	Distributed	Increasing
Attack Type (AT)	Categorical	Add	Remove	
Window Size (WS)	Ordinal	4	11	

The analysis process has three phases. Each phase performs ANOVA and the Kruskal-Wallis (K-W) test (when nonnormality is a concern) using JMP software. ANOVA was used to test if the main effects and/or interactions of the factor levels are significant. ANOVA was also used in some cases to test which factor resulted in better performance of TP% and FP%. As described in Chapter III, the null hypothesis is that their effects are not significant. This research used an alpha of 0.05, therefore a p-value from the ANOVA table that is less than 0.05 will resulted in rejection of the null hypothesis and conclusion that the treatment means were different. The analysis began by evaluating the effects of all factors and full factorial interactions. Second, the window size model parameter was evaluated. Third, the attack types were evaluated at each of the model factor levels. This process is then repeated using a modified dataset that contains no abnormal cycles and only uses the attack type of adding alarms. Using the modified dataset, the remaining attack scenario factors were evaluated.

All Factors

First, ANOVA was used to evaluate all the factors and possible interactions. Using JMP, there were five treatments with significant effects for the TP% response. There was one treatment with a significant effect for the FP% response. Table 8 (left side) shows the treatments for the TP% response and their corresponding percent

80

contribution to the total sum of squares. Table 8 (right side) shows all the significant treatments for FP%. The window size (WS) and number of cycles attacked (NC) had the highest percent contribution for the TP%. For the FP%, window size was the only significant factor and had about a 96% percent contribution. This means that the performance of the model is sensitive to the window size parameter. The residuals for both responses appeared approximately normally distributed using a normal quantile plot. The residuals plotted against the predicted value did not show any issues with variance. The next section analyzes window size alone for its impact on TP% and FP%.

Table 8. TP% and FP% top significant factors, full model.

			TP%						FP%		
Source	DF	Sum of Squares	F Ratio	Prob > F	Percent Contribution	Source	DF	Sum of Squares	F Ratio	Prob > F	Percent Contribution
WS	1	0.962	803.61	<.0001	70.35%	WS	1	0.00032	262.43	<.0001	95.77%
NC	1	0.330	275.70	<.0001	24.13%						
WS*NC	1	0.017	14.60	0.0002	1.28%						
NC*NT*AS	1	0.011	8.91	0.003	0.78%						
NC*NT*AS*AT	1	0.008	6.82	0.0094	0.60%						

Window Size

The factorial experiment included a factor that only concerned a parameter of the anomaly detection model: window size. Earlier in this chapter, when the SCADA alarm thresholds were set, it appeared that window size had a large effect on the TP% and FP%. The last section concluded that window size (WS) was the most significant factor on TP% and FP%. Here, an ANOVA was used to statistically evaluate the effect of window size and quantify its impact.

First, the effects of window size on TP% were analyzed. A graph of TP% for each window size (WS) is shown in Figure 22. The green diamonds on the graph represent the 95% confidence intervals for the mean of that treatment. This is just a

reference since the data are not normally distributed. Looking at Figure 22, the mean for

the larger window size looks much higher than the mean for the smaller window size;

though, there appears to be an overlap of some of the samples. The results of the

ANOVA indicate that a window size of four had a mean response between 60.36% and

62.90% and a window size of 11 had a mean between 83.92% and 86.64%. The larger

window size was able to detect more of the attacks and abnormal cycles, though the

results are much less than the 100% detection rate of abnormal cycles when no attacks

were present. The K-W test was also performed because the residuals appeared to

deviate from normality. With a chi-squared p-value of less than 0.0001, the null

hypothesis that the mean of the ranks of the two window sizes being identical was

rejected and the results of the ANOVA are confirmed. These results indicate that the

model performance is sensitive to the window size setting; a larger window size resulted

in better detection rates.
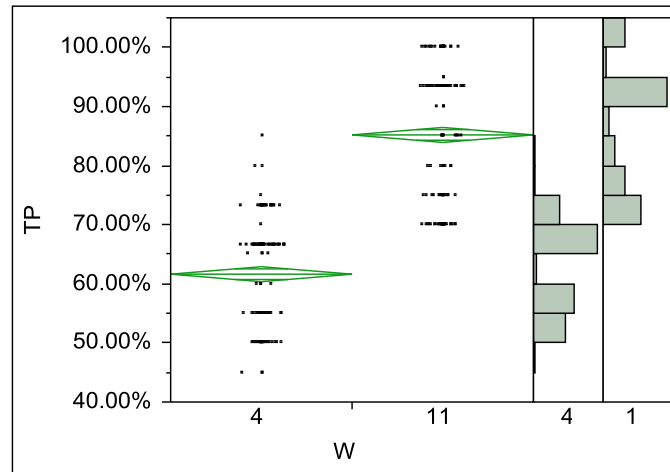
Figure 22. TP% values for small and large window size (WS).

Next, the impact of window size on FP% are analyzed. A graph of TP% for each

window size (WS) is shown in Figure 23. Here, Figure 23 shows a dramatic difference

between the FP% of each window size with the larger window size having poorer

performance; though, the scale of the graph is only between 2.4% and 3.6% FP%. The

results of the ANOVA indicate that a window size of four had a mean response between

2.80% and 2.82% FP% and a window size of 11 cycles had a mean between 3.25% and

3.28% FP%. Here, the smaller window size had better performance with a lower FP%,

though it was only a slight difference. Again, the residuals appeared to deviate from

normality. The K-W test also concluded, with a p-value less than 0.0001, that the

treatment mean ranks were different and that the smaller window size had a lower mean

rank FP%. Here, the model setting using a smaller window size resulted in fewer false

positives, but only by a small amount.
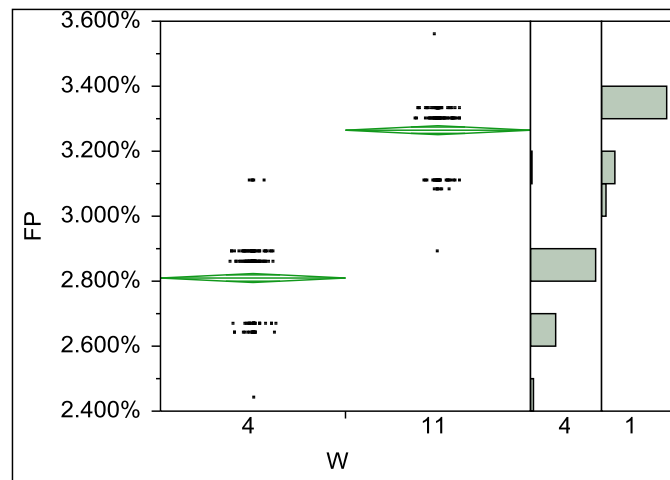
Figure 23. FP% values for small and large window size (WS).

The window size had a significant impact on the mean response for both TP% and

FP%. The larger window size was better at detecting problem cycles and the smaller

window size had a lower FP%. This is likely a result of smaller window sizes having a

larger anomaly threshold, which reports fewer false alarms but detects fewer problems.

Overall, the larger window size had an improvement in TP% between 21.02% to 26.25%

and only 0.45% to 0.48% more FP%. As discussed previously, a SCADA system owner

values problem/attack detection over lower false positives; therefore, a larger window

size would be preferred. Since the larger window size significantly outperformed the

smaller window model in detecting attacks/problems, the remaining analysis mainly

focuses on the impact of the attack scenario factors on the larger window model and the

TP% response.

Attack Scenario Factors

Now that the analysis has shown the significance of window size on TP% and

FP% overall, the impact of the attack scenario factors on the higher-performing model

will be evaluated. The data is reduced to only include the window size of 11 cycles. An

ANOVA is used to evaluate the significance of all the attack scenario factors and their

full factorial interactions.

The TP% significant factors for window size 11 are shown in Table 9. The larger

window size had three significant treatments: number of cycles attacked (NC), attack

type (AT), and the interaction between the number of targets attacked and the attack type

(NT*AT). There were no significant factors for FP%. The residuals appeared normally

distributed. The attack type is examined next.

Table 9. TP% and FP% significant treatments for window size 11.

TP%						FP%					
Source	DF	Sum of Squares	F Ratio	Prob > F	Percent Contribution	Source	DF	Sum of Squares	F Ratio	Prob > F	Percent Contribution
NC	1	0.580	482.05	<.0001	90.95%	<none>					
NT*AT	1	0.017	13.87	0.0003	2.62%						
NT	1	0.017	13.87	0.0003	2.62%						

The two levels of attack type are adding alarms and removing alarms to the

SCADA system. This research is interested to determine if it is more difficult to detect

removing alarms from a system that does not have a lot alarms occurring, such as the

water treatment data used here. It is expected that the removing alarm attack type is more

difficult to detect since anomalies are only flagged for large positive prediction errors.

An ANOVA was used to test the hypothesis that the mean TP% for adding alarms is

equal to the mean TP% for removing alarms. A graph of TP% for the attack type (AT) of

adding alarms (Add) and removing alarms (Rem) is shown in Figure 24. In Figure 24, it

appears that the model is doing better at detecting the attack type of adding alarms than

removing alarms. The results of the ANOVA indicate that adding alarms had a mean

TP% between 85.32% and 89.26% and removing alarms has a mean TP% between

81.11% and 85.05%. The results confirm that adding alarms were detected more often

than removing alarms. The K-W test also concluded, with a p-value 0.0002, that the

treatment mean ranks were different and indicated that adding alarms had a higher mean

rank TP% than removing alarms. These results indicate that the detection model is better

at detecting attacks where alarms are added versus attacks where alarms are

removed/hidden. Therefore, this model may have a difficult time detecting an attacker

changing components and hiding the changes from the operator. This makes sense since

the model only using positive changes in entropy and removing alarms would likely

decrease the entropy of the system.
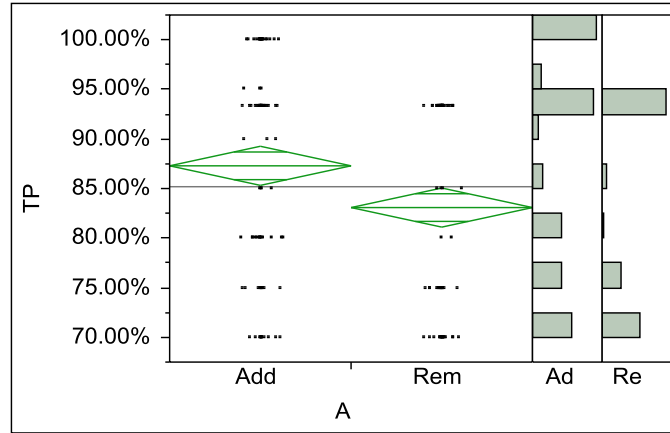
Figure 24. TP% values for the attack type (AT) of adding alarms (Add) and removing alarms (Rem).

At this point in the analysis, there is a concern that the detection of the attack scenarios is biased due to the known problems/abnormal cycles in the data. There are 14 abnormal cycles in the data that are 100% detected when no alarms are present; this could be causing the overall detection percent to be high. For example, if all 14 problems are detected and 3 attacks are not detected, the overall TP% is $\frac{14}{17} = 82.35\%$. This may give the impression that the model is detecting about 80% of problems and attacks, when in reality the model is only detecting the system problems and is missing the attacks. This may be because the model was optimized to detect the system problems. (It may be of interest in future research to optimize the detection model using a combination of known system problems and known attacks instead of known system problems alone.) The presence of abnormal cycles in the data prevented conclusions to be drawn concerning the detection of attacks alone. Therefore, a modified dataset containing no abnormal cycles was used to analyze the attack scenarios.

Factorial Experiment - Modified Dataset

The water treatment plant dataset was modified to analyze the performance of the anomaly detection model against attacks without a potential bias from abnormal cycles. This research was interested in evaluating how well the model detects the different attack scenarios without being biased by the system problems the model was optimized to detect. To create the modified dataset, every cycle with an abnormal state was removed. Additionally, the cycles before and after the abnormal cycle were also removed in case those readings were affected by the abnormal cycle. The full factorial experiment was repeated on the modified dataset, except that only the adding alarms attack type was used. Each replicate had 48 samples.

All Factors

An ANOVA was used to evaluate all the treatments in the experiment. The significant treatments are shown in Table 10. The main effects and interaction of the number of targets (NT) and the number of components (NC) were the treatments with the largest percent contribution to the TP% sum of squares. Window size was also a significant factor for TP%. For FP%, window size was the only significant factor and had about a 95% contribution to the sum of squares. The model parameter of window size will be analyzed to determine if smaller window sizes were significantly different than a larger window size when no abnormal cycles were in the data.

Table 10.   TP% and FP% significant factors, modified dataset.

TP%						**FP%**				
Source	DF	Sum of Squares	F Ratio	Prob > F	Percent Contribution	Source	DF	Sum of Squares	F Ratio	Prob > F	Percent Contribution
NT	1	4.800	85.16	<.0001	64.99%	WS	1	0.00035	242.74	<.0001	95.45%
NC*NT	1	1.611	28.59	<.0001	21.82%						
WS	1	0.300	5.32	0.0221	4.06%						
NC*NT*AS	1	0.252	4.47	0.0358	3.41%						

Window Size

The performance of window size was analyzed using the modified dataset. Figure 25 shows a graph of the TP% values for each window size (WS). The graph shows that the mean ranges slightly overlap. Using ANOVA, the mean TP% for window size four is between 21.50% and 36.56% and the mean TP% for window size 11 is between 34.41 and 49.48%. Though they overlap at the 95% confidence interval, the ANOVA results indicate that the two window sizes are statistically different at a p-value of 0.0177. Since the residuals appeared to deviate from normality, the K-W test was performed. The K-W test confirmed that the mean ranks were different at a p-value of 0.0043 and that the larger window size had a larger mean rank TP% than the smaller window size. The results indicate that the model setting using a larger window size was better at detecting attacks, as it was when using the original dataset.
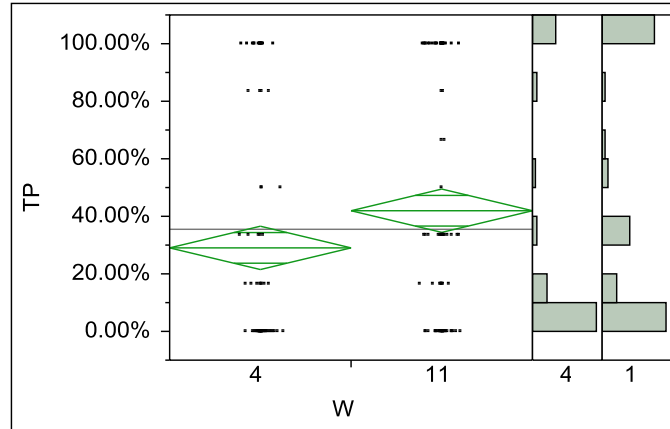
Figure 25. TP% for each window size (WS) using the modified dataset.

The same procedure was used to analyze the impact of window size on FP% for the modified dataset. Figure 26 shows a graph of the FP% values for each window size (WS); the smaller window size appears to have a better FP%, though the scale is only between 1.8% and 3.2% FP%. The ANOVA test resulted in a mean FP% between 2.30% and 2.35% for window size four and between 2.92% and 2.97% for window size 11. The K-W test confirmed the results with a p-value less than 0.0001. For both window sizes, the mean FP% was less than 3%. Similar to the results using the original data, the smaller window size setting had a better performance concerning false positives, but there is little improvement and it does not outweigh the attack detection performance of the larger window setting.
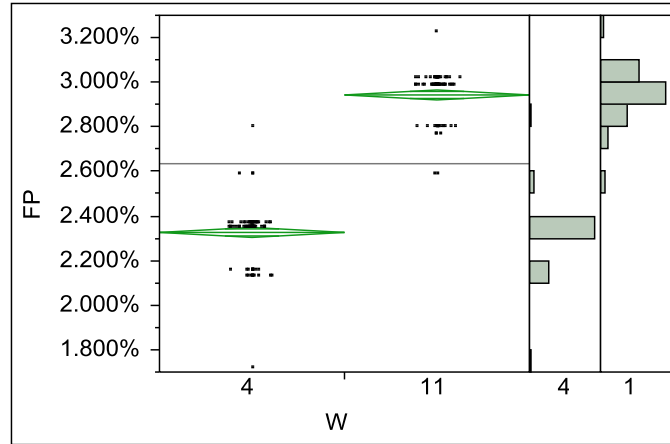
Figure 26. FP% for each window size using the modified dataset.

The results from analyzing the impact of window size on TP% were similar to the results from the unaltered dataset that contained abnormal cycles. This analysis shows that the model was detecting between 85-89% of problems and attacks overall, but only 34-49% of the attacks alone were being detected. The confidence range of the mean increased for the modified dataset, which indicates that there was more variation in the percent of attacks alone detected than the percentage of attacks and problems detected overall. Next, the significance of the attack scenario treatments were evaluated.

Attack Scenario Factors

A full factorial ANOVA was performed on each window size using the modified dataset. Here the significant treatments for both window sizes were compared. Table 11 shows the results for window size four and 11 using the modified dataset. The original data had the number of cycles (NC) as the highest contributing factor; using the modified data, it was the number of targets attacked (NT). The 11-cycle window size, shown on the right side of Table 11, also included the NC*NT interaction as a significant treatment. The four-cycle window size included the same top two treatments, plus the three-way

90

interaction of attack spacing (AS), NC, and NT. Operationally it makes sense that the

smaller window size was sensitive to the spacing of the alarm manipulations, whereas the

larger window size is not. Smaller window sizes would likely detect alarm manipulations

spaced over a short interval.

Table 11. TP% significant treatments for window size four and 11 using the modified data.

Window Size 4, Modified Data					Window Size 11, Modified Data				
Source	DF	Sum of Squares	F Ratio	Prob > F	Percent Contribution	Source	DF	Sum of Squares	F Ratio	Prob > F	Percent Contribution
NT	1	4.800	88.24	<.0001	70.21%	NT	1	4.800	82.29	<.0001	80.00%
NC*NT	1	1.611	29.62	<.0001	23.57%	NC*NT	1	0.938	16.07	0.0001	15.63%
NC*NT*AS	1	0.252	4.63	0.0339	3.69%						

The effects of the number of targets and cycles attacked, and their interaction,

were analyzed for window size 11. The results of these effects are shown in Figure 27.

Increasing the number of cycles attacked (NC) decreased the TP%; the rate of decrease in

TP% depended on the number of targets (NT) attacked (bottom left quadrant of figure).

In other words, fewer attacks were detected when there were more attacks applied. This

may be due to the entropy of the sliding window leveling off for successive cycle attacks.

Increasing the number of targets attacked (NT) increased the TP%; the rate of increase

depended on the number of cycles (NC) attacked (top right quadrant of figure). When

five components were attacked in one cycle only, the model always detected the attack.

A larger number of targets attacked mimics more components being out of threshold for a

given cycle and therefore the model is better able to detect them.
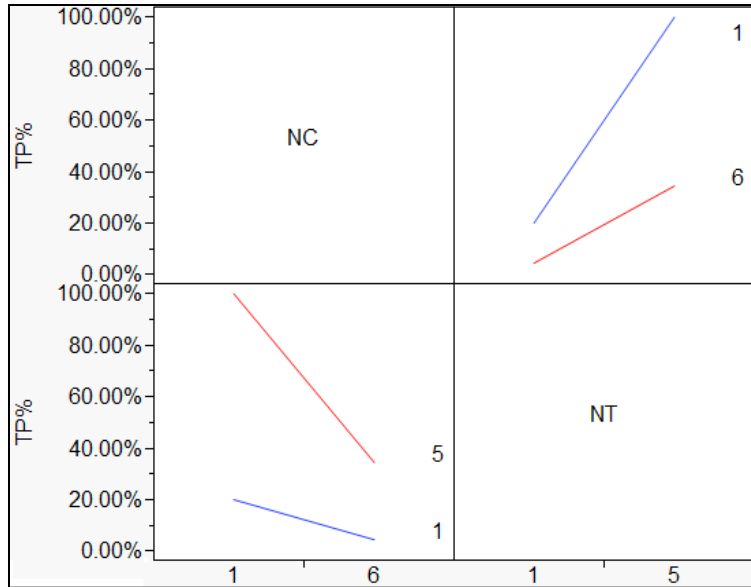
Figure 27. Effects of number of cycles (NC) and number of targets (NT) attacked for window size 11 using the modified dataset.

Attack Distribution

The attack distribution did not appear as a significant factor for the TP% for either window size. To confirm the lack of impact from using a grouped (*Grp*), distributed (*Dst*), or increasing (*Inc*) alarm manipulations, an ANOVA of the attack distribution factor on the TP% was performed. Figure 28 and Figure 29 display a graph of the TP% values for the attack distribution (AD) levels of grouped (Grp), distributed (Dst), and increasing (Inc) attacks for window size four and 11, respectively. There did not appear to be a difference in the TP% means for the levels of attack distribution in either of the graphs. With a p-value of 0.77 for both window sizes, there was not enough evidence to reject the null hypothesis that the means were equal. The K-W had the same result with a p-value of 0.99 for window size four and 0.85 for window size 11. These results indicate that the model had a similar performance for the differ types of attack distributions on this data set.
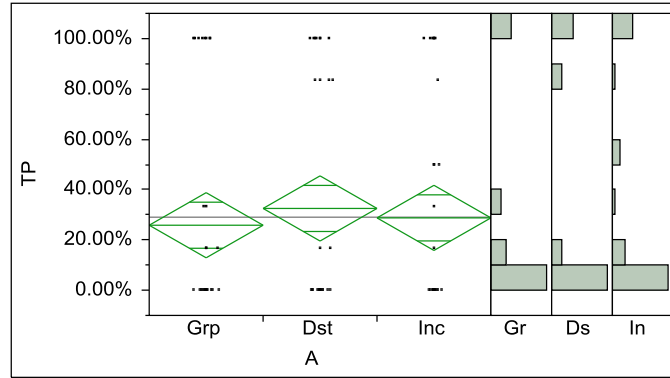
Figure 28. TP% values for the attack distribution (AD) levels of grouped (Grp), distributed (Dst), and increasing (Inc) attacks, using window size 4.
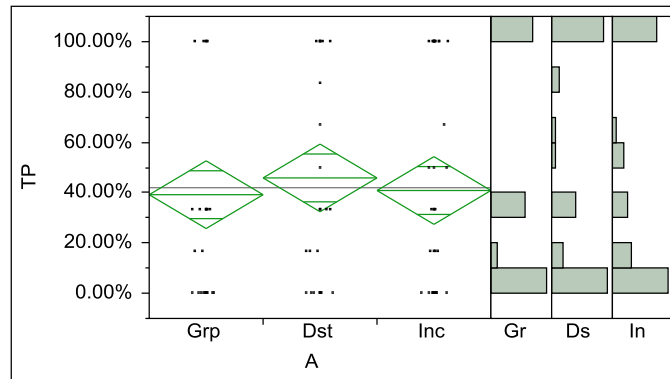
Figure 29. TP% values for the attack distribution (AD) levels of grouped (Grp), distributed (Dst), and increasing (Inc) attacks, using window size 11.

Conclusion

Overall, the entropy based anomaly detection model was successful in detecting 100% of the abnormal system states and some of the attacks using data from a water treatment plant. Higher threshold tolerances (and therefore fewer component alarms) resulted in higher detection percentages and fewer false alarms. The larger window size was significantly better at detecting attacks than the smaller window size, resulting in an overall TP% improvement of about 20%. The smaller window size was more sensitive to the attack spacing, but neither window size had attack distribution as a significant factor. The larger window size resulted in an increase in FP% of about 0.5%, which equates to

93

about two false alarm days in a year. This may be acceptable depending on the cost of responding to a false alarm. The analysis showed an interaction between the number of cycles attacked and the number of targets attacked. More attacks were detected when more components were attacked. Fewer attacks were detected when more cycles were attacked, which may be due to the sliding window causing smaller increases in entropy for successive cycles attacked.

Ultimately, the anomaly detection model performed well in detecting system problems in the water treatment plant data, which is expected since the model settings were optimized to detect all of the system problems. The detection model did not detect a majority of attacks overall, but did well against isolated attacks involving a larger number of components.

V. Conclusions

Conclusions

As critical infrastructure SCADA systems continue to use network connections to the internet and increase their vulnerability to cyber attacks, steps will be required to improve their security. This research developed an anomaly detection model based on information entropy for use on a supervisory control and data acquisition (SCADA). The objective was to create a model using SCADA component alarms to identify system problems and attacks and to analyze the impact of different attack scenarios. Entropy was used to quantify the uncertainty of the distribution of SCADA alarms over time and an anomaly flag was triggered if changes in entropy exceeded a defined threshold.

A proof of concept was demonstrated by applying the anomaly detection model to SCADA data from a water treatment plant. The results are particular to the dataset used. The model detected 100% of the known system problems at an observed false alarm percentage of 3.4%. The research found the model to be more effective when the component alarm thresholds allowed fewer alarms, though the false positive percentage increased when there were too few alarms. A full factorial experiment was conducted to analyze the significance of different attack scenario factors and the window size model parameter. A larger window size resulted in much higher true positive percentage (~20%) and only a slight increase in false alarm percentage (~0.5%). The number of cycles attacked was the significant factor with the most percent contribution to the total sum of squares. The interaction of attack type (adding or removing alarms) and the number of targets attacked was also significant. There was a concern that the results

were biased due to the abnormal cycles in the data, therefore the experiment was repeated after removing those cycles.

The model was also evaluated on detecting attacks only, without any known system problems present. The water treatment plant dataset was modified to remove all known abnormal cycles/system problems. The factorial experiment was used to analyze the model, but looking at the attack type of adding attacks only. This experiment focused on adding alarms only because the model was optimized to detect these types of attacks. The analysis revealed that only about 35-50% of the attacks were detected versus about 85-89% detection of attacks system problems combined. The most significant treatments affecting the true positive percentage were the number of targets attacked and the interaction of the number of targets attacked and the number of cycles attacked. The results showed that attacks were detected more often when more components were attacked in the same cycle. Additionally, attacking more cycles decreased the detection rate. This is likely due to the sliding window covering more than one attacked cycle and suppressing the change in entropy. The results did not show a statistical difference for changing the attack distribution. The literature review indicated that a slow and increasing attack may be harder to detect using the sliding window approach, but the results from the water treatment data, and the model parameters evaluated here, indicate otherwise.

Overall, it is recommended that this model be further evaluated to determine if it is useful for real-time anomaly detection on a live SCADA system. The results found here may be specific to this water treatment plant data set. This research indicates that the model may be able to detect system problems and certain types of alarm manipulation

attacks. It also showed that attacks involving a small number of components or attacks against successive cycles may go undetected by this model. The next sections describe some of the limitations to this research, its contributions, and potential future areas of study with this topic.

Limitations

There were some limitations to this research. The water treatment dataset was missing many sensor readings. The missing values were imputed using the last observed value. This may have adversely affected the baseline data used to set the anomaly threshold and/or impacted the false alarm percentage. Additionally, the water treatment data did not specify the alarm thresholds for each sensor. These values were derived using the anomaly detection model. Finally, the anomaly flag directed the user to a specific cycle where a problem or attack may have occurred, but did not specify which components had the issue.

Contributions

The greatest contribution from this research is a methodology for using information entropy in an anomaly detection model to detect problems in a SCADA system. The results of this thesis show a proof of concept that an entropy based anomaly detection model may be useful to SCADA operators. The model converts continuous data into discrete message types, which is necessary for applying information theory. The model is scalable to SCADA systems with more or less components, though the detection model appeared to have poorer performance when too many alarms were present. This methodology may also be useful for non-SCADA systems. The model is

independent of the time unit of the SCADA polling cycle. Here, each cycle was one day, but the model is insensitive to the unit of time.

The procedure for defining the sensor thresholds could be applied in a more general sense. This research used the anomaly detection model to set the SCADA component alarm thresholds. The process involved changing the alarm thresholds and determining the resulting detection rate for the abnormal states and the false alarm percentage. This may also be useful for setting alarm thresholds to optimize attack detection.

Future Research

The next step for evaluating this entropy based anomaly detection model is to analyze its use on other SCADA systems. The results of this research are only conclusive to the water treatment plant data used here and the attack scenarios applied. The results may differ for other systems and attack scenarios. The potential exists for applying this model for real-time anomaly detection using systems with polling cycles occurring on the scale of minutes instead of days. Future research should compare the performance of this model to other anomaly detection systems.

The model parameters are an area for future research. The model used in this research depended on the forecasting algorithm to flag anomalies. The methodology in Chapter III used the simple exponential smoothing technique, but the smoothing constant, alpha, was set to one using the dataset in Chapter IV. This resulted in the prediction using only the last observed entropy value. Other settings for alpha and the moving average interval may have different results on an alternate SCADA system.

98

The literature review for this research related several different information theoretic measures that have been used for anomaly detection. The model used in this study used information entropy. There is a potential for adapting this model to employ such measures as conditional entropy, relative entropy, mutual information, and entropy error estimation.

Finally, there may be potential to apply the anomaly detection model at the component level or using a subset of the total number of components. Some work in this area was started with disappointing results. At the individual component level, the entropy over short windows varied greatly from zero to one. This occurred because a small window covers only a few cycles and the messages often fall into one of three categories: no alarms, half alarms, or all alarms. If all the messages are the same (all alarms or no alarms), the entropy is zero. If half of the messages are alarms, the entropy is one. Using a single component and a small window size may not be advantageous for anomaly detection because the model could be too sensitive to entropy changes. Alternatively, one may be able to weigh the importance at the component level. Winter, Lampesberger, Zeilinger, and Hermann (2011) weighted the prediction errors based on the importance of the event evaluated. Similarly, an operator of a SCADA system may be able to prioritize the components of their system.

Appendix A. Using the Anomaly Detection Model

Setting Up Model with New Data

A user must use the following steps to add different data to the model to ensure

the spreadsheet functions and Visual Basic macros work properly (regions requiring user

formatting are highlighted in green):

1. Copy messages to the *Data* spreadsheet in the format shown on the sheet.
2. Complete the threshold value table as it applies to the new data on the *Data* sheet.
3. Change the *Origin* and *Alarm* message columns on the *Entropy(t)* sheet to match the number of components in the new data.
4. Change the attack matrices on the *Entropy(t)* spreadsheet to match the number of components and number of cycles in the new data.
5. Adjust all the cell and array names in the Name Manager that begin with an underscore "_" to match the appropriate number of cells and columns.
6. Copy or delete cells with formulas to the length of the columns as appropriate.

Running the Model

The below steps describe how to run the anomaly detection model:

1. Ensure the following Microsoft Excel Add-Ins are installed: "Analysis ToolPak", "Analysis ToolPak – VBA", and "Solver Add-in".
2. Enter the number of cycles considered for analysis on the *Data* sheet. If the user is running an experiment consisting of different window sizes then this number should be equal to the total number of cycles minus the largest window size. This will cause the total number of cycles evaluated for each experimental run to be the same.
3. On the *Summary* sheet, enter the known systems problems' cycle numbers and problem codes. Type any number for the attack code if you do not have attack code numbers. Leave the origin blank.
4. Set the window size and baseline on the *Entropy(t)* sheet. The baseline should not contain any system problems or attacks.
5. Press the *Calculate Baseline* button.
6. On the *Baseline* sheet, press the *Optimize* button to set the smoothing parameters.
7. Determine the results with no attacks:
    a. Return to the *Entropy(t)* sheet and press the *Clear Attacks* button to remove any attacks in the attack matrix.
    b. Press the *Calculate Attack* button.
    c. On the *Summary* sheet, press the *Detection Results* button and then *Save Output in New Sheet*.
8. Evaluate attacks:
    a. Enter attacks (by adding "1" in appropriate cycle and component on the attack matrices).

b. Press the *Calculate Attack* button.
  c. On the *Summary* sheet, press the *Detection Results* button. Next, either copy and paste the results onto the saved sheet or press *Save Output in New Sheet* to start a new sheet.
9. Run factorial experiment:
  a. On the *DOE* sheet, enter the desired levels for each continuous factor, the attack start cycle, and the number of replicates. Ensure the total number of replicates does not exceed the total possible.
  b. Press the *DOE* button.
  c. Copy and paste the results onto the saved sheet from Step 7.
  d. Return to Step 4 and repeat.

Appendix B: Detailed Description of Water Treatment Dataset

The database used in this project was found in the UCI Machine Learning

Repository (Bache & Lichman, 2013).   The water treatment process includes: input to

the plant, input to the primary settler, input to the secondary settler, outputs, and

performance measurements. Figure 30 shows a brief description of each variable. The

variables are colored to show the differences in where they are located in the water

treatment process.

```
 1 Q-E      (input flow to plant)
 2 ZN-E     (input Zinc to plant)
 3 PH-E     (input pH to plant)
 4 DBO-E    (input Biological demand of oxygen to plant)
 5 DQO-E    (input chemical demand of oxygen to plant)
 6 SS-E     (input suspended solids to plant)
 7 SSV-E    (input volatile supended solids to plant)
 8 SED-E    (input sediments to plant)
 9 COND-E   (input conductivity to plant)
10 PH-P     (input pH to primary settler)
11 DBO-P    (input Biological demand of oxygen to primary settler)
12 SS-P     (input suspended solids to primary settler)
13 SSV-P    (input volatile supended solids to primary settler)
14 SED-P    (input sediments to primary settler)
15 COND-P   (input conductivity to primary settler)
16 PH-D     (input pH to secondary settler)
17 DBO-D    (input Biological demand of oxygen to secondary settler)
18 DQO-D    (input chemical demand of oxygen to secondary settler)
19 SS-D     (input suspended solids to secondary settler)
20 SSV-D    (input volatile supended solids to secondary settler)
21 SED-D    (input sediments to secondary settler)
22 COND-D   (input conductivity to secondary settler)
23 PH-S     (output pH)
24 DBO-S    (output Biological demand of oxygen)
25 DQO-S    (output chemical demand of oxygen)
26 SS-S     (output suspended solids)
27 SSV-S    (output volatile supended solids)
28 SED-S    (output sediments)
29 COND-S   (output conductivity)
30 RD-DBO-P (performance input Biological demand of oxygen in primary settler)
31 RD-SS-P  (performance input suspended solids to primary settler)
32 RD-SED-P (performance input sediments to primary settler)
33 RD-DBO-S (performance input Biological demand of oxygen to secondary settler)
34 RD-DQO-S (performance input chemical demand of oxygen to secondary settler)
35 RD-DBO-G (global performance input Biological demand of oxygen)
36 RD-DQO-G (global performance input chemical demand of oxygen)
37 RD-SS-G  (global performance input suspended solids)
38 RD-SED-G (global performance input sediments)
```

Figure 30. The 38 sensors/variables for the water treatment dataset.

The operational classification of the plant state was provided with the dataset for each sample. Table 12 shows the 13 states and the number of samples with that class. The four largest classes are in red and represent the normal plant states. The other states comprise the abnormal states that consist of 14 samples. The abnormal states were used as known system problems in Chapter IV.

Table 12. Operational classes of plant state and number of samples with that state.

Class Description	Number of Samples
Class 1: Normal situation	275
Class 2: Secondary settler problems	1
Class 3: Secondary settler problems	1
Class 4: Secondary settler problems	4
Class 5: Normal situation with performance over the mean	116
Class 6: Solids overload	3
Class 7: Secondary settler problems	1
Class 8: Storm	1
Class 9: Normal situation with low influent	69
Class 10: Storm	1
Class 11: Normal situation	53
Class 12: Storm	1
Class 13: Solids overload	1

Appendix C: Thesis Poster

References

Arackaparambil, C., Bratus, S., Brody, J., & Shubina, A. (2010). Distributed Monitoring of Conditional Entropy for Anomaly Detection in Streams. Paper presented at the *Parallel & Distributed Processing, Workshops and PhD Forum, IEEE International Symposium*, 1-8.

Bache, K., & Lichman, M. (2013). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science.

Basharin, G. (1959). On a Statistical Estimate for the Entropy of a Sequence of Independent Random Variables. *Theory of Probability and Its Applications*, 4(3), 333-336.

Bentley Systems, Inc. (2004). SCADA/Model Integration: The Rules for Success. Whitepaper.

Chunlei, W., Lan, F., & Yiqi, D. (2010). A Simulation Environment for SCADA Security Analysis and Assessment. Paper presented at the Measuring Technology and Mechatronics Automation (ICMTMA) International Conference, 342-347.

Clarke, G., & Reynders, D. (2004). *Practical Modern SCADA protocols: DNP3, 60870.5, and Related Systems.*

Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory 2nd Edition*.

Denning, D. E. (1987). An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, (2), 222-232.

Denning, D. E. (2012). Stuxnet: What has Changed? *Future Internet*, 4(3), 672-687.

Dixon, S. R., & Wickens, C. D. (2006). Automation Reliability in Unmanned Aerial Vehicle Control: A Reliance-Compliance Model of Automation Dependence in High Workload. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(3), 474-486.

Fallier, N., Murchu, L., & Chien, E. (2011). W32.Stuxnet Dossier v1.4. Cupertino, CA: Symantec Corporation.

GAO. (2007). Government Accountability Office. Critical Infrastructure Protection: Multiple Efforts to Secure Control Systems are Under Way, but Challenges Remain (GAO-07-1036). Washington, DC.

Gelper, S., Fried, R., & Croux, C. (2010). Robust Forecasting with Exponential and Holt–Winters Smoothing. *Journal of Forecasting*, 29(3), 285-300.

Gerard, S. (2005). Safety Recommendation. Washington, DC: National Transportation Safety Board.

Harris, B. (1975). The Statistical Estimation of Entropy in the Non-parametric Case. Wisconson Univ-Madison Mathematics Research Center.

Herzel, H., & Grosse, I. (1997). Correlations in DNA Sequences: The Role of Protein Coding Segments. *Physical Review E*, 55(1), 800.

Hunter, J. S. (1986). The Exponentially Weighted Moving Average. *Journal of Quality Technology*, 18(4), 203-210.

Igure, V. M., Laughter, S. A., & Williams, R. D. (2006). Security Issues in SCADA Networks. *Computers & Security*, 25(7), 498-506.

Jung, S., Song, J., & Kim, S. (2008). Design on SCADA Test-bed and Security Device. *International Journal of Multimedia and Ubiquitous Engineering*, 3(4), 75-86.

Kahler, B. (2013). Development and Evaluation of a Graph-based Intrusion Detection System. Paper presented at the ARC, 283-286.

Krishnamurthy, B., Sen, S., Zhang, Y., & Chen, Y. (2003). Sketch-based Change Eetection: Methods, Evaluation, and Applications. Paper presented at the *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, 234-247.

Kutner, M. H., Nachtsheim, C., & Neter, J. (2004). *Applied Linear Regression Models (4th ed.)*. Boston: McGraw-Hill/Irwin.

Lakhina, A., Crovella, M., & Diot, C. (2005). Mining Anomalies Using Traffic Feature Distributions. Paper presented at the *ACM SIGCOMM Computer Communication Review*, 35(4) 217-228.

Lee, W., & Xiang, D. (2001). Information-Theoretic Measures for Anomaly Detection. Paper presented at the *IEEE Security and Privacy Symposium*, 130-143.

Linda, O., Vollmer, T., & Manic, M. (2009, June). Neural Network Based Intrusion Detection System for Critical Infrastructures. In *Proceedings of International Joint Conference on Neural Networks*,1827-1834.

Luce, R. D. (2003). Whatever Happened to Information Theory in Psychology? *Review of General Psychology*, 7(2), 183-188.

McConahey, J. (2012, January 3). Using Modbus for Process Control, Automation. *Control Engineering.*

Montgomery, D. (2013). *Design and Analysis of Experiments (8th ed.)*. New Jersey: John Wiley & Sons, Inc.

Mumaw, R. J., Roth, E. M., Vicente, K. J., & Burns, C. M. (2000). There is More to Monitoring a Nuclear Power Plant than Meets the Eye. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 42(1), 36-55.

National Transportation Safety Board. (2010). Pipeline Accident Report. (No. NTSB/PAR-12/01 PB2012-916501). Washington, D.C.: National Technical Information Service.

Nychis, G., Sekar, V., Andersen, D. G., Kim, H., & Zhang, H. (2008). An Empirical Evaluation of Entropy-based Traffic Anomaly Detection. Paper presented at the *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, 151-156.

Obama, B. (2013). Executive Order 13636: Improving Critical Infrastructure Cybersecurity. The White House.

Roberts, S. (1959). Control Chart Tests Based on Geometric Moving Averages. *Technometrics*, 1(3), 239-250.

Roulston, M. S. (1999). Estimating the Errors on Measured Entropy and Mutual Information. *Physica D: Nonlinear Phenomena*, 125(3), 285-294.

Sanger, D. E. (2011, June 1, 2012). Obama Order Sped Up Wave of Cyberattacks Against Iran. *New York Times*, pp. A1.

Schutzer, D. (1982). Concepts and Thoughts Concerning Control Strategy for Conducting Information Warfare. Paper presented at the *5th MIT/ONR Workshop on C3 Systems*, 12.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27, 379-423.

Shannon, C. E. (1956). The Bandwagon. *Institute of Radio Engineers Transactions on Information Theory*, 2(1), 3.

Shaw, W. T. (2006). *Cybersecurity for SCADA Systems*. Pennwell books.

Simply Modbus. (2013). Operation Manual - Simply Modbus Slave. Retrieved Oct 23, 2013, from http://www.simplymodbus.ca/RTUslavemanual2.htm.

Stouffer, K., Falco, J., & Scarfone, K. (2008). Guide to Industrial Control Systems (ICS) Security. *National Institute of Standards and Technology Special Publication*, 800(82).

Su, R., & Yurcik, W. (2005). A Survey and Comparison of Human Monitoring of Complex Networks. University of Illinois at Urbana-Champaign: National Center for Advanced Secure Systems Research.

U.S. Department of Homeland Security. (2011). Common Cybersecurity Vulnerabilities in Industrial Control Systems. Industrial Control Systems Cyber Emergency Response Team.

Verdu, S. (1998). Fifty Years of Shannon Theory. *IEEE Transactions on Information Theory*, 44(6), 2057-2078.

Wagner, A., & Plattner, B. (2005). Entropy Based Worm and Anomaly Detection in Fast IP Networks. Paper presented at the *14th IEEE International Workshops On Enabling Technologies: Infrastructure for Collaborative Enterprise*, 172-177.

Wang, E., & Liu, K. (2009). A Human Factors Improvement on Supervisory Alarms in Wastewater Treatment System. Paper presented at the *Industrial Engineering and Engineering Management International Conference,* 618-621.

Wang, Y., Zhang, Z., Guo, L., & Li, S. (2011). Using Entropy to Classify Traffic More Deeply. Paper presented at the *Sixth IEEE International Conference on Networking, Architecture, and Storage*, 45-52.

Weidling, D. (2000). City of Arcata Water Supply. Retrieved Oct 28, 2013, from http://www.sonic.net/~dlw2/modbus.jpg.

Winter, P., Lampesberger, H., Zeilinger, M., & Hermann, E. (2011). On Detecting Abrupt Changes in Network Entropy Time Series. Paper presented at the *Communications and Multimedia Security Lecture Notes in Computer Science* (7025), 194-205.

Xia, T., Qu, G., Hariri, S., & Yousif, M. (2005). An Efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm. Paper presented at the *24th IEEE International Performance, Computing, and Communications Conference*, 11-17.

Yang, D., Usynin, A., & Hines, J. W. (2006). Anomaly-Based Intrusion Detection for SCADA Systems. In *5th Intl. Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies*, 12-16.

Young, W., Weckman, G., & Holland, W. (2010). A Survey of Methodologies for the Treatment of Missing Values within Datasets: Limitations and Benefits. *Theoretical Issues in Ergonomics Science*, 12(1), 15-43.

Zhang, L., Qin, X., Wang, Z., & Liang, W. (2009). Designing a Reliable Leak Detection System for West Products Pipeline. *Journal of Loss Prevention in the Process Industries*, 22(6), 981-989.

Zhu, B., & Sastry, S. (2010). SCADA-Specific Intrusion Detection/Prevention Systems: A Survey and Taxonomy. Paper presented at the *Proceedings of the 1st Workshop on Secure Control Systems*, Royal Institute of Technology (KTH), Stockholm, Sweden. 77-92.

Vita

Major Jesse G. Wales attended the University of Texas at Austin, from which he received a bachelor of science in physics in May 2003. After receiving a commission as a second lieutenant from Officer Training School, he worked in the Air Force Research Laboratory at Wright-Patterson AFB, OH, from 2003 to 2006 performing night vision and cockpit display research. During that time, Major Wales earned a graduate certificate in Operational Technology from the Air Force Institute of Technology.

Major Wales' second assignment was at Officer Training School, Maxwell AFB, AL. From 2006 until 2009, he served as an instructor in the 24th Training Squadron, a flight commander in the 22nd Training Support Squadron, and an assistant director of operations for the 24th Training Squadron. In 2008, he was a top-third graduate from Squadron Officer School.

In 2009, he was assigned to the Air Force Operational Test and Evaluation Center, Detachment 4, Peterson AFB, CO. He performed operational test planning, execution, and analysis for Air Force space systems. He served as a lead test analyst and a test director. Here, Major Wales earned a graduate certificate in Space Systems Management from the University of Colorado at Colorado Springs.

In August 2012, Major Wales was competitively selected to study Operations Research at the Air Force Institute of Technology, Graduate School of Engineering and Management. There he was inducted into the Tau Beta Pi and Omega Rho honors societies. Upon graduation, he will be assigned to the Air Force Inspection Agency at Kirtland AFB, NM.

			Form Approved
REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 27-03-2014	2. REPORT TYPE Master's Thesis	3. DATES COVERED *(From – To)* October 2012 – March 2014
TITLE AND SUBTITLE Analysis of a SCADA System Anomaly Detection Model Based on Information Entropy		**5a. CONTRACT NUMBER**
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Wales, Jesse G., Major, USAF		**5d. PROJECT NUMBER**
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865		**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENS-14-M-32
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally left blank		**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/RHIQ (example)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
  Distribution statement A. Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States

14. ABSTRACT

SCADA (supervisory control and data acquisition) systems monitor and control many different types of critical infrastructure such as power, water, transportation, and pipelines. These once isolated systems are increasingly being connected to the internet to improve operations, which creates vulnerabilities to attacks. A SCADA operator receives automated alarms concerning system components operating out of normal thresholds. These alarms are susceptible to manipulation by an attacker. This research uses information theory to build an anomaly detection model that quantifies the uncertainty of the system based on alarm message frequency. Several attack scenarios are statistically analyzed for their significance including someone injecting false alarms or hiding alarms. This research evaluates the use of information theory for anomaly detection and the impact of different attack scenarios.

15. SUBJECT TERMS
SCADA, entropy, information theory, alarm, attack, anomaly detection

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Jennifer L. Geffre, Maj, USAF, AFIT/ENS
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	124	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636, ext 4646 Jennifer.Geffre@afit.edu
U	U	U			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18